Dynamical System Approach to Insider Threat Detection

Nitin Kanaskar*, Jiang Bian†, Remzi Seker*, Mais Nijim‡ and Nuri Yilmazer‡

*Department of Computer Science
University of Arkansas at Little Rock
Little Rock, Arkansas 72204
Email: nvkanaskar|rxseker@ualr.edu
†Department of IT Research
University of Arkansas For Medical Sciences
Little Rock, Arkansas 72204
Email: jbian@uams.edu
‡Department of Electrical Engineering and Computer Science
Texas A&M Kingsville
Kingsville, Texas
Email: mais.nijim|nuri.yilmazer@tamuk.edu

Abstract—Insider attacks have the potential to inflict severe damage to an organizations reputation, intellectual property and financial assets. The primary difference between the external intrusions and the insider intrusions is that an insider wields power of knowledge about the information system resources, their environment, policies. We present an approach to detecting abnormal behavior of an insider by applying Dynamical System Theory to the insiders computer usage pattern. This is because abnormal system usage pattern is one of the necessary precursors to actual execution of an attack. A base profile of system usage pattern for an insider is created via applying dynamical system theory measures. A continuous monitoring of the insiders system usage and its comparison with this base profile is performed to identify considerable deviations. A sample system usage in terms of application system calls is collected, analyzed, and graphical results of the analysis are presented. Our results indicate that dynamical system theory has the potential of detecting suspicious insider behavior occurring prior to the actual attack execution.

I. INTRODUCTION

Organizations are increasingly using and relying on wide scale information system networks because of changing business requirements like mergers, collaborations, and acquisitions. As a result, the backbone of their sustainability information systems is constantly being faced with external threats like malicious hackers and insiders. Out of the two, insider threat obviously has greater damage capability than the other one. An insider is usually a disgruntled employee, a terminated employee or a person who illegitimately acquires valid employee credentials. The damage caused by an insider can range from introduction of virus, worms, Trojans into the organizations networks to theft of intellectual property or money and destruction of vital data for the organization. Cappelli [1] describe some of the indicators/observations about employee behavior in their case studies which can be used by organization to increase their monitoring efforts. Many times, this type of employee behavior is characterized by unusual system command execution for installing a Trojan, virus or a worm, privilege escalation attempt. Thus, this abnormal behavior for a user is a critical signal which should raise alarms of the security mechanisms deployed in the organization network. This necessity drives the need to implement and maintain a continuous employee monitoring system wherein employees system usage pattern is stored to create a profile specific to the employee. Any activity on the employees part which significantly deviates from this base profile should raise alarm.

This monitoring system based on employees system usage profile brings us to the topic of anomaly detection technique for insider threat detection. We apply Dynamical System Theory to characterize a given users system usage pattern for an insider is created via applying dynamical system theory measures. A continuous monitoring of the insiders system usage and its comparison with this base profile is performed to identify considerable deviations. A sample system usage in terms of application system calls is collected, analyzed, and graphical results of the analysis are presented. Our results indicate that dynamical system theory has the potential of detecting suspicious insider behavior occurring prior to the actual attack execution.

II. BACKGROUND

As noted by Moore [3] the motivation for the insider to launch attack on organization’s information resources can be psychological, social, and personal in nature. They explain typical observations that can be made about the insiders behavior in the organization which act as strong indicators of insider threat issue. Zhang [4] came up with a Hierarchical Active Defense Model and Framework for detecting insider threats. Anderson [5] proposed System Dynamics model to capture activities and processes involving insider and their interaction with the organizational elements. They tried to model complex interrelationships between policies, trust in employees, employee expectations, and an insiders motivation to launch an attack.

Liu [6] compared different features like n-grams, frequencies of system calls, and their parameters and sought to detect outliers by employing Bays k nearest neighbor algorithm. They
found system call parameters to be more sensitive than the other two feature representations. We explain the Dynamical system theory in the next section.

III. DYNAMICAL SYSTEM APPROACH

Dynamical system theory has been explored by researchers to understand and explain nonlinear behavior in nature like turbulence in sea, atmosphere, fluctuation in wildlife populations, accumulation of vehicles on highways, oil flow in underground pipes, electronic devices and many other universally diverse events [7]. An excellent introduction to nonlinear dynamics and chaos can be found in [8]. Researchers have performed extensive analysis as to how stock prices and foreign exchange rates vary from nonlinear dynamics theory viewpoint [9].

A Dynamical System, put simplistically, traverses a set of states called state space apparently in a random fashion. Its behavior appears random to standard statistical tests, although it traverses the state space on the basis of some deterministic rule. System states can be defined in terms of values of variables in the system. As an example, the traffic flow on highway when observed, appears random in movement. But it is governed by deterministic rules laid down by traffic system in the given place. Three variables in the system that can be observed are:

- Number of vehicles stopped at a traffic light,
- Number of vehicles moving between this traffic light and neighboring traffic lights, and
- Average speed of vehicles moving between neighboring traffic lights.

These variables are affected by unpredictable events like accidents, existence of an ambulance in traffic, or just individual driving styles. Hence, this Dynamical system behavior has some randomness or unpredictability along with some deterministic rules.

In the similar vein, a users system usage patterns can be perceived as traversing a set of states. Depending upon the users role in an organization, her access control rights to resources are determined. Accordingly commands executed by the user on a system in organization network can be utilized to form a usage pattern. This usage pattern is associated to system calls invoked by applications executed by the user and user profile is generated from system call trace analysis. Determinism of the system comes from the fact that instructions of application code are executed sequentially. Thus, instructions are executed in a predefined sequence, but depending upon input commands to the application by the user, this sequence may change. System call sequences represent individual actions performed by an application like opening a file, closing a file, writing to memory buffer, reading data from disk, so on executed on behalf of user. Sequence of system calls invoked by the application can be considered as sequence of states it goes through during its life cycle. In other words, system call trace becomes system observable to represent its behavior. System call sequence for the application can be observed and studied over a period of time to characterize a users system usage pattern in terms of degree of determinism. Figure 1 presents our overall approach for using an application’s system call sequence as a dynamical system observable. System call trace generated by the application program may be considered as an observable time series from which we can reconstruct the state space of the application dynamics through a process called Embedding which is explained in section below describing Approximate Entropy and its theoretical background can be referred to from Takens [10].

We believe an application’s long term behavior and its users behavior in terms of degree of determinism can be better understood by reconstructing the application’s state space through system call trace and then applying certain dynamical system analysis tools. We choose Approximate Entropy [11], Central Tendency Measure, and Recurrence Plots analysis techniques to study and characterize application’s long term behavior. These measures define a system’s characterization in terms of degree of determinism, similarity and rate of variability which we find useful for creating a users normal behavior profile.

We explain the theory behind these Dynamical system measures which form the basis of our system usage characterization tools.

A. Approximate Entropy

Using this measure, we study an applications behavior from the perspective of the systems information complexity and utilize the state space reconstructed for the application from its system call time series data. The Approximate Entropy measure was proposed by Pincus [11] to assess a systems information complexity. It is a statistical measure capable of classifying complex systems with relatively few data points. Approximate Entropy has been successfully utilized to quantify complexity in physical systems as well as physiological systems. It works satisfactorily on small lengths of time series data to give their complexity measure [12]. Consider a one dimensional time series:

$$ x = x(1), x(2), x(3), ..., x(N) \quad (1) $$

All the scalar components of this time series are equi-spaced in time. A series of m-dimensional points...
where $N_0 = \text{Number of } u(j) \text{ such that Euclidean distance } |u(i) - u(j)| < r$, $r$ is the radius of the sphere in $m$ dimensional space centered at $u(i)$. Euclidean distance is given as

$$|u(i) - u(j)| = \sqrt{\sum_{k=0}^{m-1} [x(i + k) - x(j + k)]^2} \quad (4)$$

We define

$$C_i^m(r) = (N - m + 1)^{-1} \sum_{i=1}^{N-m+1} C_i^m(r) \quad (5)$$

$$\phi^m(r) = (N - m + 1)^{-1} \sum_{i=1}^{N-m+1} \log C_i^m(r) \quad (6)$$

Approximate Entropy for some fixed values of $m$ and $r$ is defined as

$$ApEn(m, r) = \lim_{N \to \infty} \left[ \phi^m(r) - \phi^{m+1}(r) \right] \quad (7)$$

**B. Central Tendency Measure**

Here, we attempt to establish some measure of the chaotic behavior in the applications system call time series by calculating its central tendency measure (CTM) [13]. CTM is a metric to evaluate the degree of variability in a given data. CTM has been employed in the analysis of various physiological processes like heart rate variability and behavior of schizophrenic patients [14]. Second order difference plot for a time series $a(1), a(2), ..., a(n)$ obtained by plotting $a(n + 2) - a(n + 1)$ Vs. $a(n + 1) - a(n) - a(n)$ acts as a tool to measure this variability factor for a dynamical system. In a time series $a(1), a(2), ..., a(n), a(n + 1), a(n + 2)$ of length $N$, if $r$ is denoted as the radius of the sphere around the origin, then

$$CTM = \left[ \frac{\sum_{i=1}^{N-2} \delta(d_i)}{(N - 2)} \right] \quad (8)$$

where

$$\delta(d_i) = \begin{cases} 1, & \text{if } |a(i + 2) - a(i + 1)|^2 - |a(i + 1) - a(i)|^2 < r^2, \\ 0, & \text{Otherwise} \end{cases}$$

The value of radius $r$ is selected depending upon the nature of data [14].

**C. Recurrence Plots**

This is a dynamical time series analysis technique developed by J. P. Eckmann [15]. It graphically demonstrates time correlation between different points on the state space of a dynamical system. Point $(i,j)$ in a Recurrence Plot is marked black (or 1) if two points representing the system states at instants $i$ and $j$ are close enough as defined by a criterion of Euclidean distance $r$. Thus,

$$RP(i,j) = \begin{cases} 1, & \text{if } d(x(i),x(j)) \leq r, \\ 0, & \text{Otherwise} \end{cases}$$

Points $x(i)$ and $x(j)$ are part of an embedded time series data. The $i$th row of this multidimensional vector represents the system state at $ith$ instant. Recurrence Plots have been exploited to discern hidden patterns and nonstationeries in time series data for physiological systems. Different structural elements in Recurrence Plots denote certain qualitative aspects of the time series data in terms of determinism, recurrent patterns. We define Percent Recurrence, Percent Determinism, and Percent Ratio [16] to characterize an application’s normal behavior.

1) **Percent Recurrence**: Percent Recurrence gives the fraction of points in multidimensional state space repeating previous system dynamics. It helps us distinguish a process with periodic dynamic behavior from that with an aperiodic behavior. More the points that are observed at same states of a dynamical system, the more periodicity exhibited by its reconstructed state space.

2) **Percent Determinism**: Percent Determinism is associated with line structures present in a Recurrence Plot. A Line of Identity (LOI) is formed on a plot for all points where $i = j$. This is the line having the slope of one which passes through the origin and divides plot area into two congruent triangles. There may appear other line structure(s) which are parallel to LOI. Such a line in the plot is formed by points (1s) that are diagonally adjacent with no white spaces (0s) in between. For example, if pairs of consecutive points $[x(i),x(j)], [x(i+1),x(j+1)], [x(i+2),x(j+2)], ..., [x(i+N),x(j+N)]$ in multidimensional state space of a dynamical system exhibit the same dynamics, then the corresponding points placed in the Recurrence Plot form a line parallel to the LOI. Percentage of points in these lines articulate how much structure of the state space repeats on consecutive points of the state.

3) **Percent Ratio**: Percent Ratio is the ratio of Percent Determinism to Percent Recurrence in the plot. This quantity captures the extent to which the system state space is experiencing sudden variations. So it is an effective indicator of sudden transitions in state dynamics of a process. All of the
above defined Recurrence Plot parameters strongly highlight presence of hidden rhythms, and determinism characteristics in data.

IV. Experimental Setup

We have performed a simulation experiment which mimics the user commands during the attack execution or just prior to attack execution. To represent normal valid user system usage we run a vi text editor session on ubuntu Linux machine. Then to represent abnormal system usage by user, two commands ls and rm are run from within vi session. These commands spawn 2 different processes. System call traces invoked by all the 3 processes are captured separately using strace utility on Linux. Traces for abnormal processes (ls and rm) are mixed with the one for normal process (vi). The normal trace for vi and trace mixed with abnormal processes are pre-processed to convert into one dimensional number time series by using system call number mapping from unistd.h. These one dimensional time series are subjected to process of embedding. Dynamical system measures are computed for cumulatively increasing lengths of these embedded time series. These values are plotted for normal and the abnormal processes on the same plot. The dotted curve corresponds to abnormal process and continuous line curve corresponds to normal process trace.

V. Analysis Approach

From the one dimensional time series, the state space of daemon’s behavior is reconstructed by the process of embedding as described in the Approximate Entropy section. Embedding, captures the time evolution of the system dynamics by reconstructing the state space with specific embedding dimension and time delay values. Analysis approach employed in our work is referred to as Non-Clustered Subsystem approach. In this approach each system call in a trace is treated as a single data point with a unique number. There is another approach where all system calls belonging to certain functionality - file i/o, disk i/o, memory allocation, checking access rights - can be represented with a common unique number corresponding to that functionality which is termed as Clustered approach. Subsystem refers to each child process created by a parent process for the application. The trace of each child process is collected separately and treated as a subsystem of the main system corresponding to parent process. Each child process trace is subjected to the process of embedding and then graphical analysis utilizing Dynamical system analysis tools. Embedding dimension value is varied between 2 to 15 and embedding delay is varied between 1 and 3. We performed experiments with 4 approaches - a)non-clustered subsystem, b)clustered subsystem, c)non-clustered aggregate, d)clustered aggregate. Aggregate approach consists of combining each child process trace of the application and subjecting this 2 dimensional matrix to dynamical system analysis tools. Amongst the four approaches, it is observed that non-clustered subsystem approach with embedding dimension \( m = 2 \) and embedding delay \( \tau = 1 \) renders the best system dynamics approximation. Differences between two curves of Dynamical System measures for normal and abnormal processes are best projected for these values of embedding dimension and delay. Our graphical results for these parameter values are presented in the next section.

VI. Results and Observations

Our simulation experiment consists of a normal Linux process session of vi editor and 2 abnormal processes. Our objective is to identify sudden deviations in user behavior by analyzing the differences between dynamical system measures for the normal process and the 2 abnormal processes. Typically, in a real scenario normal and abnormal commands are going to be mixed by an insider. Hence we collect small lengths of the 2 abnormal processes system call trace and mix them with the normal process trace at known intervals. For these normal and abnormal traces, dynamical system measures are plotted on graphs and compared. Figure 2 shows plots for the selected 5 Dynamical System measures Percent Recurrence, Percent Determinism, Percent Ratio, Central Tendency Measure, and Approximate Entropy for normal process and normal process mixed with abnormal process after first 200 data points.

One of the goals of our experiment is to identify minimum data length required to be analyzed to detect abnormal user session. Hence, the abnormal process trace is mixed with the normal trace at increasing indices. Figure 3 below shows Dynamical system measures for the normal trace and abnormal trace when abnormal trace is mixed with the normal after 400 data points.

Figure 4, Figure 5 below show Dynamical system measures for the normal trace and abnormal trace when abnormal trace is mixed with the normal after 700 and 1100 data points respectively.

Our experiment consists of running two different abnormal processes corresponding to Linux commands - ls and rm in the background of normal process of vi editor. Now, graphical
Dynamical system measures are shown considerably. We vary the data length of 400 onwards, the differences in all the traces start showing up after data length of 200. But, until the data length becomes 400, these differences are not considerable. We infer that the best differentiation and hence superior detection capability is obtained at m=2 and τ=1.

**VII. DISCUSSION AND CONCLUSION**

We have tested the Dynamical System approach to identify abnormal user system usage for insider threat detection on Ubuntu Linux with Kernel version 2.6.32-21-generic. Normal user session is represented in terms of system call trace from a vi editor session. Abnormal user session is represented in terms of two other system call traces captured by running ls and rm commands from within vi session. Total data length considered in our analysis is 2000. Matlab scripts which are used to compute Dynamical system measures give the output graphs within 20-30 seconds, do not cause much system load in terms of processor cycles or memory.

After analyzing 400 data points of embedded time series

<table>
<thead>
<tr>
<th>N Size</th>
<th>Percent Recurrence</th>
<th>Percent Determinism</th>
<th>Approximate Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>1000</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>1500</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>2000</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Fig. 3. vi Normal trace and abnormal trace mixed after 400 normal data points
Fig. 4. vi Normal trace and abnormal trace mixed after 700 normal data points
Fig. 5. vi Normal trace and abnormal trace mixed after 1100 normal data points
Fig. 6. vi Normal trace and abnormal (rm) trace mixed after 200 normal data points

Analysis of Dynamical system measures is presented for vi trace and rm process trace mixed with vi trace. Figure 6, Figure 7, Figure 8, Figure 9 display dynamical system measures when abnormal trace is mixed with normal trace after 200, 400, 700, and 1100 data points respectively.

From all the above graphical plots, it is observed that the difference between the dynamical system measures for the normal system call trace (vi session) and the abnormal traces start showing up after data length of 200. But, until the data length becomes 400, these differences are not substantial and not all measures show the difference. With the data length of 400 onwards, the differences in all the Dynamical system measures are shown considerably. We vary the system parameters embedding dimension m from 2 to 15 and embedding delay τ from 1 to 3. It is observed that the difference between the Dynamical system measures for embedded time series of normal and abnormal traces decreases with increasing values of both the parameters. Hence we infer that the best differentiation and hence superior detection capability is obtained at m=2 and τ=1.

**VII. DISCUSSION AND CONCLUSION**

We have tested the Dynamical System approach to identify abnormal user system usage for insider threat detection on Ubuntu Linux with Kernel version 2.6.32-21-generic. Normal user session is represented in terms of system call trace from a vi editor session. Abnormal user session is represented in terms of two other system call traces captured by running ls and rm commands from within vi session. Total data length considered in our analysis is 2000. Matlab scripts which are used to compute Dynamical system measures give the output graphs within 20-30 seconds, do not cause much system load in terms of processor cycles or memory.

After analyzing 400 data points of embedded time series
of both normal and abnormal data with Dynamical system measures, the difference is exhibited in the graphical results. A collective analysis of the dynamical system measures of information complexity, degree of determinism, and periodicity can be used to characterize a given user system usage pattern. This characterization of user normal behavior further can be used to detect suspicious behavior. Our results in terms of detecting abnormal system dynamics from the normal ones are promising for the simulation experiment we performed. Although system call sequence analysis is pivotal to identifying anomalous behaviors of an application, it is the minimum length of the trace to be analyzed that is the key to decide whether it is an anomalous sequence. Through this paper, we have successfully demonstrated that Dynamical System approach applied to system call sequences can be leveraged to detect abnormal deviations in system usage by a user for the purpose of insider threat detection.

**Fig. 7.** vi Normal trace and abnormal (rm) trace mixed after 400 normal data points

**Fig. 8.** vi Normal trace and abnormal (rm) trace mixed after 700 normal data points

**Fig. 9.** vi Normal trace and abnormal (rm) trace mixed after 1100 normal data points

**References**


