A Security Model for Detecting Suspicious Patterns in Physical Environment

Simon Fong, Zhuang Yan
Faculty of Science and Technology, University of Macau
ccfong@umac.mo

Abstract

In the view of escalating global threat in security, it is imperative to have an automated detection system that can pick up suspicious patterns of human movement in physical environments. It can give a forewarning before a planned attack happens or an ultimate security is breached. In the past, significant research on the Intrusion Detection was established, but limited to virtual environments like computer networks and operating systems. In this paper, we proposed a general security model for detecting suspicious patterns in physical environment. Suspicious patterns are subtle and we showed that they can be detected via an experiment.

1. Introduction

Security for physical environment usually and basically relies on physical locking mechanisms and CCTV surveillances. These security measures would independently guard at individual entries and cover a certain area of the whole compound. This common security scheme works fine for enforcing access control and challenges to the users. By basic principle a user who possesses an access token of authority and/or is being recognized as a legitimate identity such as biometric, is granted an entry upon an entry through a door or moving across certain area. These prevalent measures may satisfy most of the security requirements today, but may not meet the future escalating security threats that requires forewarning.

In the post 9/11 world, there is much focus on connecting the dots in both virtual and physical environments. Intrusion Detection System (IDS) is a mature technology that detects intrusion by monitoring activities in several aspects of the network or operating systems, piecing scattered information together for some insights [1]. Likewise, emails could be traced and related to the user profiles for modeling behavior [2]. A lot of research effort has been devoted to intrusion detection on virtual platform. However relatively little work is on intrusion detection in physical environment. Some pioneers are [3] and [4] who developed logical models for detecting suspicious patterns in contact-based smart cards and contactless RFID cards in physical access environments respectively. Tamas in [5] developed profiles that describe user behavior in Computer Forensic investigations. These works although in a somewhat early stage, have shed some light on modeling user behavior especially the abnormal ones in a physical environment. For instance, [3] defined a real-time detection model for inspecting irregular access patterns of users movements.

In this paper, we argue that by connecting multiple access reference points, we can gain a better understanding of the user's behavior than a single entry validation verdict such as access "Granted" or "Denied". Over time, the user's patterns can be learnt and the system is able to tell whether a new trail is normal or suspicious. A user who possesses legitimate access right traveled certain areas by certain fashion may be deemed absolutely normal in the traditional access control system. However in the context of detecting anomaly for physical environment security, certain legal behaviors when combined with the preceding and subsequent actions plus other factors, and put under test, we could discover subtle suspicious elements in the eye of a domain expert. For example, a technician may have access rights to all the staff rooms in the office. But if he appears to be repeatedly visiting certain rooms, especially after his normal working hours, his trail is considered to be suspicious. For another example, if the technician's movement patterns seems to be always within the close proximity of a VIP staff where his role is not a body-guard, or one user keeps following another user, this kind of behaviors is wary. Such suspicious patterns that are subtle in nature would be easily eluded from the current physical security system. In this paper, our research aim is to tackle the problem of how to pick out this category of 'suspicious' patterns in a physical environment by computing technology.

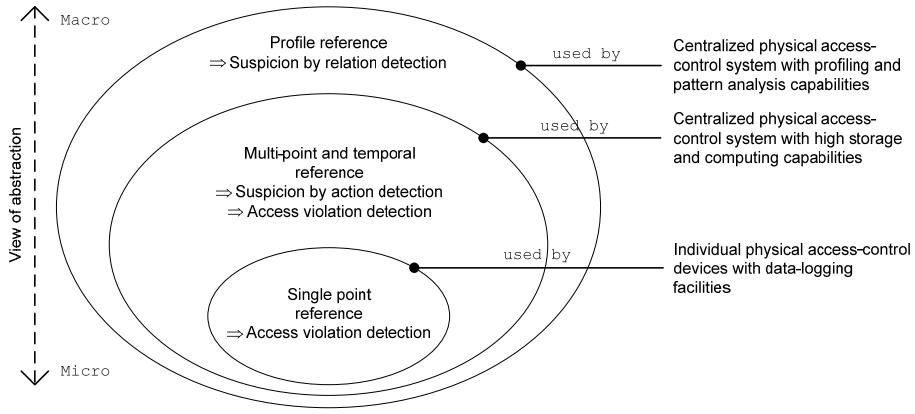

Figure 1 Scopes of reference points and types of detections in a physical environment

The contribution of this paper is in twofold. First it defines a logical model that is generic and based on a simple door-lock security environment. The model is for representing physical trail data in numerical format; hence data mining and other analysis techniques can be applied on. In the data representation model, different levels of views and various types of detection can be facilitated. The second contribution is presenting the appropriate data analysis techniques for detection both misuse violation and suspicious patterns.

2. Suspicion Detection Model

The term suspicion usually means something is deviated from the norm. In here we are referring to user activities and actions that are reflected by their movement. While most security systems are capable to detect misuse or access violation, suspicious movement which are of subjective in nature remain difficult to define and to detect. We attempted to propose a model that allows rules for checking out suspicious behavior be defined, as well as a set of algorithms that can automatically detect them. Firstly we take a data-centric point of view and consider intrusion detection as a data analysis process. We suppose that intrusion in physical environment includes staged attacks, instant break-in's and a combination of both. It is believed that certain tell-tale signs can be observed, before a staged attack happens such as planning, plotting and spying etc, Even when the attack was being carried out, the process may contain some abnormal signs. Anomaly detection is to pick out the signs that show deviation from the normal. Misuse detection usually is to identify a single or a series of instant break-in's.

In our model, that is based on the scope views in Figure 1, anomaly detection is about identifying the abnormal usage patterns from the audit data, whereas misuse detection is about encoding and matching the intrusion patterns using the audit data, as well as monitoring for any immediate violation.

The central theme of our model is to provide a comprehensive view of the meanings of the data by single and multiple reference access points. These reference points are the fundamental elements from which we can derive and used to analyze whether the audit data contain any suspicious movement. Our model also defines a data transformation procedure that converts audit data that extracted from the logging system to abstract patterns that can readily resemble the behavior of intrusions and normal activities.

2.1. Assumptions

The primary assumptions of suspicious detection in our model are: user activities are always observable user movements through controlled check points, for example, via logging of door access and auditing mechanisms; and each user is required to use his own access card or biometric feature that can prove his identity to access through all the doors at all times. Suspicion detection in physical environment includes these essential elements:

- Every door must have installed this checkpoint access feature and every access record is logged and sent to the centralized server without failure and without any substantial delay;
- Whenever a door opens by presenting the card to the sensor, the user indeed passes through the door – moves from one area to another area. There is also no shoulder surfing;
- Both sides of the door have a separate card reader installed, so it can indicate the direction of door access. For example, we can derive from our logs to tell if a user is entering or leaving a room.

Table 1 Types of detection and what do they check in a physical environment

Hardware System	**Type of Detection**	**Possible techniques**
Individual PAC devices with simple data-logging facilities	Access violation Check access on the spot: - PIN - Access rights - Validation of card - Number of retries	Rule checking, Predicate logic, Access Control List
Centralized PAC system with high storage and computing capabilities	Access violation, Suspicion by action (micro view), c.f. [6] Check for multi-point and temporal violations: - Out of sequence - Displacement - Overstay - Number of retries Check for suspicious behavior by instant actions: The combination of the following contribute to a suspicious behavior - User ID ⇒ user access level - Location information - Times/date - Frequency - Length of stay	Color Petri-Net, Graph Traversing
Centralized PAC system with profiling and pattern analysis capabilities	Suspicion by relation (macro view) Check for suspicious behavior by relations: - Deviate from his own normal pattern in profile - Deviate from normal pattern of his cohorts - Similar patterns to other users who may be in the same or other groups - The user is being followed by other users - The user is probably being ambushed - Unusual behavior relative to other activities - Collaborative behavior among other users	Data mining (e.g. association rules, sequence pattern matching, etc.)

Ticket ID	Reader ID	*UserID*	*Timestamp*	*From_where*	*To_where*	*Access_point*	*Status*

Figure 2 Typical format of an access log recorded at the door access point; (grey areas are used in mining)

2.2. Data Preprocessing

We consider that user access patterns can be correspondingly represented by door access patterns in this tightly access control environment. In the context of monitoring user access trails, we concern information of which doors the users have passed through at which period of time, at what frequency and the semantics of the patterns in terms of having visited multiple reference points and timestamps. Let v be a vector that comprised of the following six information: a_0 = *UserID*, a_1 = *Timestamp*, a_2 = *From_where*, a_3 = *To_where*, a_4 = *Access_point*, a_5 = *Status*, such that $v_i = \{a_{i0}, a_{i1}, a_{i2}, a_{i3}, a_{i4}, a_{i5}\}$ at the i step where i is an atomic interval stepping from 0 to n.

$Trace_{k,j} : v_i \; \forall \; i = [0..n]$ where j is an unique record identifier that refers to an instance of a period of activities belongs to the user of *UserID k*. In short, we call the user just as $user_k$.

The record j starts and stops at a definite interval whose conditions can be configured by the system administrator. The conditions could be either relating to a specific time or to a significant access point. The starting condition, for example, can be at 0800 every morning and/or entering the main door of a building. Exiting certain doors likewise can be set as the ending condition.

The following are some example traces that demonstrate how they could be transformed by mapping via some predefined lookup tables.

A user with an *UserID k* has an instance of movement trace covering the segment of time period that is indexed by *j*.
$\Rightarrow Trace_{k,j} : \{v_0, v_1, \ldots v_n\}_{k,j}$

Expanding the vector v, it shows the attribute name as follow
$\Rightarrow Trace_{k,j} : \{(user_k, t_0, Room\_Num_0, Room\_Num_0, Door\_Num_0, Status_0), (user_k, t_1, Room\_Num_1, Room\_Num_1, Door\_Num_1, Status_1), \ldots (user_k, t_n, Room\_Num_n, Room\_Num_n, Door\_Num_n, Status_n)\}_{k,j}$

For an example, a typical record would have the attribute values as follow
$\Rightarrow Trace_{k,j}$:{(1003807, 20070321085642, A01, A02, 315, 1), (1003807, 20070321085702, A02, A05, 317, 1), .. (1003807, 20070321180523, A02, A01, 315, 1)}

Numerical attribute values are suitable for data-mining and other computation, however should one wants to view the record content in a readable form it is possible to do data-mapping and transform the attributes *From_where*, *To_where* and *Status* plus the time information *i* and *i*-1, to another action higher level action item. For example:
$\Rightarrow Trace_{k,j}$:{(Mr James Smith/Senior Manager, March 21, 2007, 08:56:42am, Entering through the main door from front-yard to lobby, Ok), (Mr James Smith/Senior Manager, March 21, 2007, 08:57:02am, From main lobby to own office in 20 seconds, Ok), .. (Mr James Smith/Senior Manager, March 21, 2007, 06:05:23pm, Exit through the main door from lobby to front-yard, Ok)}

Further abstraction on the data by lookup tables gives a simple format that is suitable for association rules mining
$\Rightarrow Trace_{k,j}$:{(User *k* came to work at ZZZZ hour), (User *k* visited XXX for YYYY seconds at ZZZZ hour), (User *k* visited XXX for YYYY seconds at ZZZZ hour), (User *k* visited XXX for YYYY seconds at ZZZZ hour), (User *k* visited XXX for YYYY seconds at ZZZZ hour), (User *k* finished work at ZZZZ hour)} → Normal, 87%, 79%

The lookup tables need to be predefined and updated whenever the physical layout has changed. The procedure of generating the abstract access traces (AAT) is described as follow. Briefly, each file of the trace data has two columns of integers, the first is the user IDs and the second is the access action "numbers". These numbers are indices into a lookup table of access action names. For example, the number "7" represents an access action of walking into the software engineering laboratory on a weekday morning. Some example set of traces include:

Normal traces: an AAT of the 'usual' activities of *user k* such as coming to office in the morning, and a concatenation of several other actions generated by *user k* that fall within his job scope.

Suspicious traces: an AAT of *user k* staying after normal working hour; an AAT of *user k* frequently loitering in some sensitive areas; an AAT of *user k* following another user who has no relation at work or social association; an AAT of *user k* correlates to other AAT of sensitive events, etc.

Table 2 is an example of the labeled AAT sequences. It should be noted that a suspicious trace may contain many normal sequences in addition to the abnormal sequences since the illegal activities only occur in some places within a trace

Table 2 Example of labeled AAT

AAT Sequences	Class Labels
7 2 5 6 2 8 2 9 2 7	"normal"
. . .	. . .
7 7 7 9 12 1 38 2 43	"abnormal"
. . .	. . .

Another step of pre-processing is to transform the format of one data source to another should they were extracted from different physical environments that may have different physical layouts.

2.3. Suspicion Detection

Suspicion detection consists of first establishing the normal behavior profiles for the users, and observing the actual activities as extracted from the access logs to ultimately detect any significant deviations from these profiles. Our model adopted SRI's NIDES [7] technique, for implementing a user's profile that has a set of statistical measures. To compute the deviations from the profile, NIDES uses a weighted combining function to sum up the abnormality values of the measures. The profiles are also updated periodically (i.e. aged) based on the (new) observed user behavior to account for normal shifts in user behavior (e.g., when a project deadline approaches most people tend to work late and frequent at the workshop).

Theoretically, suspicion detection technique can detect unknown suspicion pattern since they require no apriori knowledge about specific attack. Statistical-based approaches also have the added advantage of being adaptive to evolving user behaviors since updating the statistical measures is relatively easy.

However, it is often very difficult to classify a single event by a user as normal or abnormal because the unpredictable nature of most people. A user's actions during a working day or even months needs to be studied as a whole to determine whether he or she is behaving normally.

2.3.1. Mining frequent AAT. Assume every door captures the access data of the users. The user activities can be viewed in addition to several more dimensions, such as time of access, frequency and roles of the users.

For example, a technician may have legitimate access rights to the directors' meeting room. But it would be abnormal if a technician has repeatedly visited the director's boardroom room. Likewise, time plays a factor into considering whether an action is suspicious or not. An user who visited a place that is beyond his normal working hour or job scope in general would be thought suspicious. Table 3 describes some example consistent behavior of the simulated users for anomaly analysis.

Table 3. User Description

User	Normal Activities
Manager	Meeting in conference room; Working in own office
Clerk	Working in departmental cubicles;
Technician	Working in workshop; Working out stations
Cleaner	Washrooms; Cleaning every accessible area of the building

We first preprocess the raw audit logs to AAT by adding semantics into them as described in section 2.2. This usually needs certain domain knowledge. Based on the sequences of door accesses and the information of areas, raw access data are transformed into meaningful AAT. We further pre-processed the timestamps with *am*, *pm* and *night*, and kept only the abstract meaning of the door access action. The AAT records were used for user anomaly detection.

One approach by association rules is to mine the frequent patterns from the AAT data, and merge or add the patterns into an aggregate set to form the normal usage profile of a user. A new pattern can be merged with an old pattern if they have the same left-hand-sides and right-hand-sides, their support values and confidence values are both highly graded.

To analyze a user activity session, we mine the frequent patterns from the sequence of accesses during this session. This new pattern set is compared with the profile pattern set and a similarity score is assigned.

Assume that the new set has n patterns and among them, there are m patterns that have "matches" (i.e. rules that they can be merged with) in the profile pattern set, then the similarity score is simply m/n. Obviously, a higher similarity score means a higher likelihood that the user's behavior agrees with his or her historical profile.

2.3.2. Checks against historical profiles. Once each user's profile history is established, to evaluate if a user activity trail is suspicious could be a matter of checking how much this activity pattern deviates from his norm statistics in his profile. The idea of profiling can be extended from Individual Profiling (Intra-check) to Group Profiling (Inter-check). Group profiling is a task of deriving the common activity patterns in terms of statistics over AAT of users who have the same job titles or cohorts who are supposed to be doing the same things. The following suggests what kinds of checks can be facilitated upon individual and group profiles.

Individual Profiling (Intra)
- Captures and stores daily activities of a person that belongs to a group with a stereotyped job function;
- Has a set of average patterns of what considered as "normal";
- Can check against (intra) his behavior against his only daily norm profile, check if today's behavior is normal or deviate from his normal pattern;
- Check how much does his behavior conform to his norm and how much conform to the designated pattern;
- Check behavior of this user in relative to users from his same group, and users from the other groups.

Group Profiling (Inter)
- Requires domain knowledge of the job functions;
- Subject to adjustment/fine-tuning in time to come; re-grouping is possible for users over time or role changed;
- Used for checking if an individual that belongs to this group conform or deviate from the "norm" of the group;
- Group profiling can be hierarchical.

Profile checking still has a number of drawbacks before accurate insight can be drawing from its mining results. The most basic level of check as studied so far is limited to tracing the users' trails of their whereabouts. That is, only the temporal, location and frequency statistics etc are available so far. These provide limited information unless costly domain knowledge in applied on defining the meaning of each combination of temporal, proximity and access frequency elements. Quite often, we do not know finer details of what exactly they were doing. E.g. user x staying in Room A for 2 hours is all we know. Only when more details are available, a better picture/semantics of what is happening can be yielded.

2.4. Violation Detection

STAT [8], is one of the approaches we adopted that uses state transition analysis for violation detection. It represents and detects known penetration scenarios using state transition diagrams. The intuition behind this approach is that any penetration is essentially a sequence of actions that leads the user movement from an initial normal state to a compromised state. In our model, however there is only a limited number of "allowable" valid transitions and states for a physical environment, we deem a new trace that can match one of the routes in the state transition diagram "Okay", and "Violated" otherwise. Here a state transition diagram is a list of assertions in terms of allowable routes and user access privileges. A transition is labeled by a door through which a user has the right to access, and a state is a specific area of the building. The key advantage of this state transition diagram approach is that the system can accurately and efficiently determine if a sequence of access is legitimate or not. Some example traces based on Fig. 2. are given below:

A valid user trace

$$0 \rightarrow^{a} 1 \rightarrow^{f} 6 \rightarrow^{i} 7 \rightarrow^{i} 6 \rightarrow^{f} 1 \rightarrow^{d} 4$$

An invalid user trace

$$0 \rightarrow^{a} 1 \rightarrow^{f} 6 \rightarrow^{i} 7 \rightarrow^{h} 2 \rightarrow^{f} 1 \rightarrow^{d} 4$$

A valid but suspicious trace

$$0 \rightarrow^{a} 1 \rightarrow^{c} 3 \rightarrow^{h} 2 \rightarrow^{b} 1 \rightarrow^{c} 3 \rightarrow^{h} 2$$

3. Conclusion

In this paper we defined a logical security model that is generic and based on a simple door-lock security environment. The model is for representing physical trail data in numerical format; hence data mining and other analysis techniques can be applied on. In the data representation, different levels of views and various types of detection can be facilitated. In suspicion detection, normal behavior of user activities are first summarized into normal profiles, which are then flagged as probable abnormal. For violation detection, a state transition analysis is suggested that can verify if a user access pattern conforms to the legitimate graphs.

4. References

[1] Xiaodong Zhu; Zhiqiu Huang; Hang Zhou, "Design of a Multi-agent Based Intelligent Intrusion Detection System", 2006 1st International Symposium on Pervasive Computing and Applications, Aug. 2006 pp.290 - 295

[2] Salvatore J. Stolfo, Shlomo Hershkop, Chia-Wei Hu, Wei-Jen Li, Olivier Nimeskern, Ke Wang, "Behavior-based modeling and its application to Email analysis", ACM Transactions on Internet Technology, ACM Press, May 2006.

[3] Leong, A.; Fong, S.; Siu, S, "Smart card-based irregular access patterns detection system", EEE '04. 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service, 2004, 28-31 March 2004, pp.546-553

[4] Pengfan Yan, Robert P. Biuk-Aghai, Simon Fong, Yain-Whar Si, "Detection of Suspicious Patterns in Secure Physical Environments", IEEE International Conference in Information Technology and Applications (ICITA2007), January 15-18, 2007, Harbin, China

[5] Tamas Abraham, "Event sequence mining to develop profiles for computer forensic investigation purposes", Proceedings of the 2006 Australasian workshops on Grid computing and e-research - Volume 54 ACSW Frontiers '06, January 2006

[6] A. Leong, S. Fong, "Physical Control Access With Smart Intrusion Tracking And Hunting Agent", Proceeding of 2nd International conference on Intelligent Computing & Information Systems, ACM Press, Cairo, 5-7 March 2005, pp.443-451

[7] Alfonso Valdes, Anderson, Lunt, Javits and Tamaru, "Detecting Unusual Program Behavior Using the Statistical Components of NIDES", http://www.csl.sri.com/papers/5sri, May 1995

[8] S.T. Eckmann and G. Vigna and R.A. Kemmerer, " STATL: An Attack Language for State-based Intrusion Detection", Proceedings of the 1st ACM Workshop on Intrusion Detection Systems, Athens, Greece, November 2000

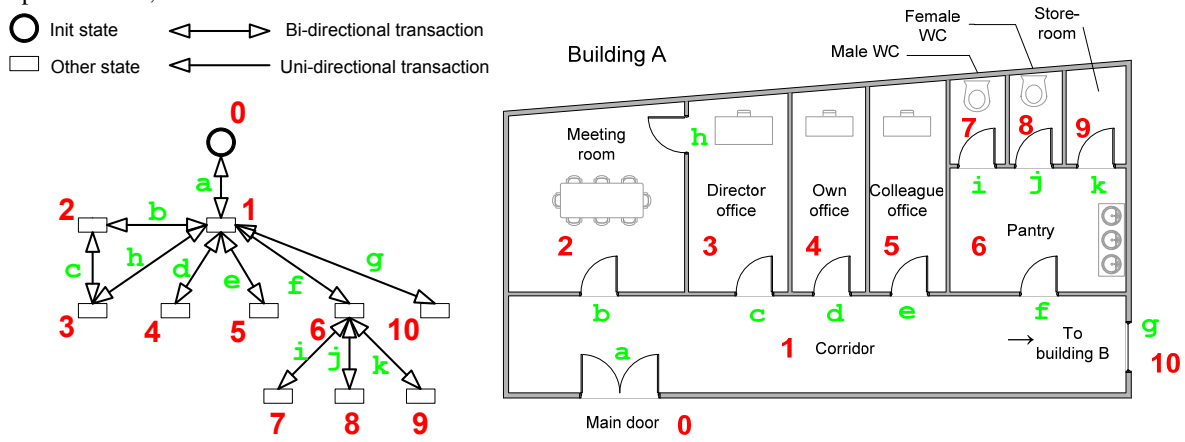

Figure 2 Example of a building layout and its corresponding Stat-Transition Diagram for violation detection