Data-Driven Model-Based Detection of Malicious Insiders via Physical Access Logs

Carmen Cheh[✉1], Binbin Chen[2], William G. Temple[2], and William H. Sanders[1]

[1] University of Illinois, Urbana, IL 61801 USA,
{cheh2,whs}@illinois.edu
[2] Advanced Digital Sciences Center, Singapore,
{binbin.chen,william.t}@adsc.com.sg

Abstract. The risk posed by insider threats has usually been approached by analyzing the behavior of users solely in the cyber domain. In this paper, we show the viability of using physical movement logs, collected via a building access control system, together with an understanding of the layout of the building housing the system's assets, to detect malicious insider behavior that manifests itself in the physical domain. In particular, we propose a systematic framework that uses contextual knowledge about the system and its users, learned from historical data gathered from a building access control system, to select suitable models for representing movement behavior. We then explore the online usage of the learned models, together with knowledge about the layout of the building being monitored, to detect malicious insider behavior. Finally, we show the effectiveness of the developed framework using real-life data traces of user movement in railway transit stations.

Keywords: Physical Access · Physical Movement · Cyber-physical Systems · Insider Threat · Intrusion Detection · User Behavior

1 Introduction

Insider threats are a top concern of all organizations because they are common and can have severe consequences. However, insider threats are very difficult to detect, since the adversary already has physical and cyber access to the organization's assets. Much state-of-the-art research [1] and many state-of-the-practice tools [2, 3] focus on the cyber aspect of insider attacks by analyzing the user's cyber footprint (e.g., logins and file accesses). However, the strength of an organization's defense mechanisms is only as strong as its weakest link. By failing to consider the physical aspect of users' behavior, an organization not only leaves itself unable to detect precursor physical behavior that could facilitate future cyber attacks, but also opens itself up to less tech-savvy attacks such as vandalism and theft [4].

Thus, physical security plays a crucial role in an organization's overall defense posture. This is especially true for critical infrastructure systems such as power grids and transportation systems in which a physical breach can have major real-world effects. Building access controls [5] are often used to limit the areas that

users can access based on their role in the organization; this is normally achieved through a relatively static assignment of a set of locations to the user's tracking device (e.g., RFID tag or access card). When a user moves between spaces (e.g., swiping a card at a door), information about this movement is logged.

Although building access control restricts the spaces that a user is able to access, it is merely the first step towards physical security. As with other access control solutions, it faces the same problem of being overly permissive [6]. But denying access to rarely accessed rooms is a costly solution, as it places the burden on administrators to grant every access request, which can lead to severe consequences, especially in time-critical situations (e.g., maintenance). Even with a restrictive set of granted permissions, the access control solutions in place do not take into account the context of a user's access.

Thus, we focus on detecting abnormalities in a user's movement within an organization's buildings. Specifically, we explore how physical access logs collected from a railway transit system can be used to develop a more advanced behavior-monitoring capability for the purposes of detecting abnormalities in a user's movement. In particular, we aim to determine 1) the feasibility of characterizing the movement behavior of users in a complex real-world system, 2) the techniques that can be applied to this detection problem, and 3) the ability to integrate real-time detection into physical security.

We provide a systematic approach to tackle these issues in a way that can be generalized to a diverse set of systems. We observe that since an organization consists of users who have a diverse set of roles, the movement patterns of users in different roles may vary vastly because of their job needs. Instead of proposing a single technique to model all users, we construct a methodical approach that selects the appropriate model based on the context of the organizational role and learns that model from historical data. More specifically, we propose metrics to determine the feasibility of modeling the behavior of certain users in a system. We then construct a model that factors in contextual information such as time and location, and show that the model can be used in an online manner. This study is supported by a set of real-life physical access traces that we collected from our industrial collaborator.

In summary, our contributions in this paper are as follows:

- We show that abnormal movement of users can be detected from physical access logs, thus strengthening a system's physical security.
- We define a framework that characterizes a user's physical movement behavior and learns models of the user's behavior using historical data.
- We evaluate our framework using real-world physical access data obtained from railway transit stations. We show that our metric properly differentiates users, allowing us to use appropriate models of user movement behavior to obtain good false positive and false negative detection rates. We also show the feasibility of performing detection in an online manner.

The structure of the paper is as follows. In Section 2, we discuss related work in the domain of anomaly detection of physical movement. Section 3 introduces our case study of railway transit systems, and Section 4 describes our framework

for detecting malicious insiders, applying it to the case study as an example. The evaluation results are presented in Section 5. Finally, future work is summarized in Section 6, and the conclusion is given in Section 7.

2 Related Work

In this section, we discuss the related work spanning domains from physical movement tracking and prediction to anomaly detection of physical movement and cyber events.

There has been a substantial amount of work on use of cyber logs (e.g., network flows and system logs) to profile users and detect events of interest. For example, Kent et al. proposed *authentication graphs* [7] to profile user behavior and detect threats using computer authentication logs in an enterprise network. In contrast, our work focuses on physical access logs, where physical-world factors, like space and time, directly impact the correlation among different access events. Despite these differences, we also observe the importance of distinguishing different user roles.

In the area of physical access control, there has been work in the route anomaly detection area that looked at people or objects moving in a geographical space that was not delineated by rooms [8–10]. Pallotta et al. [8] and Radon et al. [9] both detect deviations in the trajectory of a vessel in the maritime domain. Their approaches use contextual information, such as the speed of the vessel and weather information, in order to predict the next location of a vessel. However, in the maritime domain, the source and destination of the vessel are already known beforehand, and the anomalies are assumed to arise from the differences in trajectories. This is unlike our work, in which we focus on an indoor setting that has unpredictable destinations for each user.

Dash et al. [10] use mobile data to predict the movement of people in a geographical region. They construct multiple Dynamic Bayesian network models, each of which includes different granularities of context (e.g., day of the week vs. time of day). They predict the next visited location by analyzing the results obtained from each of those models. Unlike their completely data-driven approach of applying all models before computing the best result, we propose a more guided approach by first choosing the appropriate model based on an understanding of a person's past movement data.

In contrast to the work described above, we consider the more restrictive setting of indoor location tracking, which reduces the amount of noise in the data and allows us to identify a user's location with more confidence. Because of physical barriers that prevent a user from moving uninhibited from one space to another, the paths that a user can take are also limited.

For indoor physical access, there has been work in both movement prediction and anomaly detection. In the movement prediction domain, Gellert et al. [11] use *Hidden Markov Models* (HMMs) to predict a user's next location. They use real-world physical access data of four users from a single floor of an office building, although the size and topology of the building are very small. Their

results show that a simple Markov model of order 1 gives the best performance. Koehler et al. [12] expand on Gellert's work by using ensemble classifiers to predict how long a user will stay at a given location.

In the anomaly detection domain, different techniques to detect differences in a user's movement have been proposed. Graph models have been studied by Eberle [13] and Davis [14]. Eberle et al. [13] detect structural anomalies by extracting common subgraph movement patterns [15]. However, they only consider simplified physical layouts and do not distinguish among different user roles. Davis et al. [14] search labeled graphs for both structural and numeric anomalies and apply their approach to physical access logs in an office building.

Other models, ranging from finite state machines to specific rules, have also been studied. Liu et al. [16] model the normal movements of devices as transitions in finite state machines. Unlike us, they focus on the movement of devices (instead of people) in a hospital setting, where their main goal is to detect missing-device events. Biuk-Aghai et al. [17] focus on suspicious behavioral patterns, including temporal, repetitive, displacement, and out-of-sequence patterns. These patterns only involve the time interval between movements and the reachability of locations rather than the sequence of locations that were visited.

Finally, patents from IBM [18] and Honeywell [19] present the general design of using physical access data to detect potential security incidents. However, they do not discuss detailed designs for dealing with complicated building topology and user roles, and do not provide experimental studies on real-world traces.

3 Motivating Use Case

Physical security is of high priority for industrial control facilities and critical infrastructures. Through a project partnership, we have gained deep knowledge about the physical access control challenges faced by railway transit system operators. We will use this real-world use case to motivate our study.

Background The railway transit system is an important component of a nation's transportation system. The impact of an attack or fault in the system can be very severe, ranging from loss of service and station blackouts to derailment. For example, a Polish teenager rewired a remote control to communicate with the wireless switch junctions, causing derailment of a train and injury of twelve people [20]. Since the track was accessible by the public, the attack was easily performed. However, in our case study, the underground railway system presents a stronger barrier against such an attack. Potential loss of revenue and human life motivates the need for both physical and cyber security of such systems. In particular, the insider threat is of the utmost importance, as can be seen in the 2006 case in which two traffic engineers hacked into a Los Angeles signal system, causing major traffic disruption [21].

System Architecture A railway station consists of a single building that may house one or multiple railway lines through it. The general public accesses the railway lines by passing through fare gates in the concourse area and moving

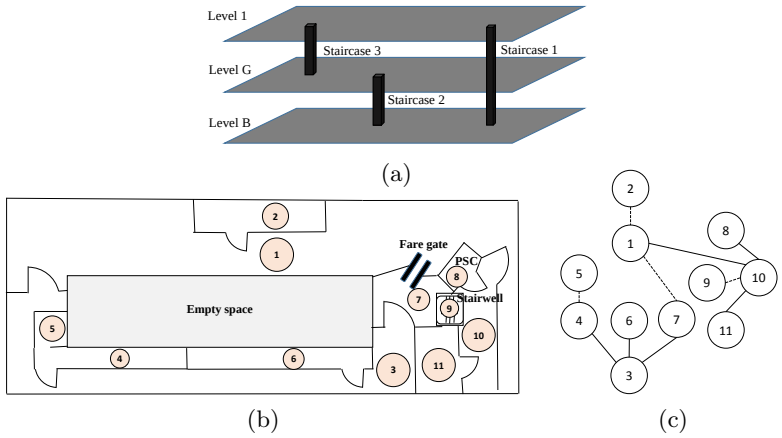


Fig. 1. (a) The different levels of a railway station building with staircases connecting two or more levels. (b) A small sample floor plan of one of the levels. The PSC room represents the Passenger Service Center. (c) Graph representation of (b). Each edge in the graph represents a pair of directed edges between the vertices. Bolded edges imply that a card reader exists on the door bordering the spaces (vertices).

to the platform. Figure 1 depicts the topology of the railway station in our case study. In addition to the concourse and platform area, the railway station contains many rooms hidden from the public eye that house the equipment necessary to maintain the running of the station and its portion of the railway track. Each room serves a specific function, and there are multiple rooms that share the same function. The rooms are distributed throughout the station on multiple levels. The railway staff can access those spaces only by swiping their access cards at readers on the doors. Although most of the doors inside the staff-only spaces have card readers, there are a number of doors that allow free access. Different stations have different floor plans, and the number of rooms within a station may vary. However, all the stations share the same types of rooms (e.g., power supply room).

Threat Model Our threat model focuses on users who have gained physical access to the rooms in a railway station. Those users may be malicious railway staff or an outsider who has gained control over an employee's access control device. Since building access control solutions are in place, we assume the adversary's goal is to tamper with devices in a room to which he or she already has physical access. In a railway station, almost all the rooms house critical assets. Thus, we cannot narrow our focus to any specific portion of the railway station to reduce the space of possible movement trajectories. The level of risk involved in letting an adversary achieve his or her goal is too high. However, restricting a user's access to rooms in a station can also result in severe consequences. Since railway staff require access to rooms in order to conduct maintenance on devices within those rooms, denying them access could cause disruption of service.

Opportunities and Challenges Unlike an enterprise system for which the office building has a simple, systematic layout across all levels (e.g., a single

corridor branching out to multiple rooms), a railway station has a complex non-symmetrical layout. There are multiple paths with varying lengths that a user can take to get from one room to another. This implies that topology is an important factor in determining whether a user's physical movement is anomalous. In addition, the railway transit system consists of diverse user roles (e.g., station operators and power maintenance staff). The job scopes of such users vary in terms of work shifts, responsibilities, and work locations, all of which affect their physical movement behavior. Even users in the same role exhibit different movement behaviors based on their assigned duties and personal habits.

The building access control system that is in place offers a limited view of users' physical movement. Since card readers may fail and certain doors are not outfitted with card readers, we are unable to determine a user's full movement trajectory. A user may also tailgate another user, and thus the access will remain invisible to us. Therefore, it is challenging to detect deviations in a user's movement behavior. We tackle this problem in the next section by integrating knowledge of the system layout and by learning models of users' behaviors from historical physical access data.

Envisioned Monitoring Currently, a railway system staff member would need to look through the physical access logs manually in order to detect malicious behavior. We aim to reduce the amount of manual effort by automatically presenting a smaller subset of potentially malicious physical accesses in real-time to the staff member. The staff member can then focus his or her attention on the smaller subset, using video surveillance to corroborate evidence of malicious activity. To aid the decision-making, we can also supplement the suspicious accesses with a model of the users' normal behavior.

4 Malicious Insider Detection Framework

In this section, we describe our framework that systematically analyzes users' physical movement logs to detect malicious insiders. The framework dissects the problem into three parts: understanding the characteristics of users' behaviors, learning a suitable model representation, and using the model together with knowledge of the system layout to estimate the probability of an abnormal access.

4.1 Preliminaries and Definitions

We define a system $sys = \{U, Env\}$ (e.g., enterprise organization, critical infrastructure) as the collection of users U who work for it, and the environment Env that contains the system's assets. The environment Env consists of both the physical and cyber aspects of the system. The physical aspect is composed of the building $\mathbf{B}$ and the physical assets $\mathbf{Q}$ within it. The cyber aspect consists of the networked computer system and its digital assets. The cyber and physical aspects are interrelated, but we focus only on the physical aspect in this paper.

We represent the building topology $\mathbf{B}$ as a directed graph $G = (V, E)$ in which the set of vertices represents the spaces S in the building. A directed edge

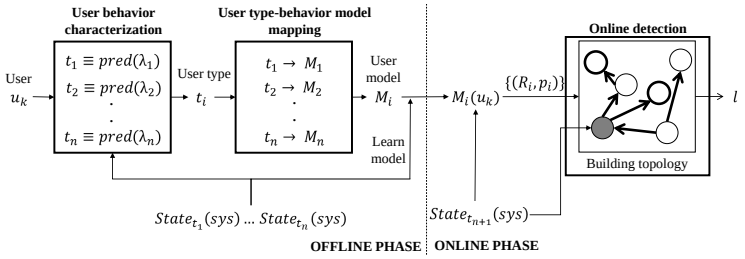


Fig. 2. The framework is divided into offline and online phases, where the offline phase is fed into the online phase.

$e(v_1, v_2)$ represents possible movement from v_1 to v_2.[3] For example, the floor plan in Figure 1b is represented as the graph in Figure 1c. The set of spaces S can be divided into two covering disjoint subsets: rooms $\mathbf{R}$, and common areas $\mathbf{C}$ (e.g., staircases, corridors). The edges have attributes that represent the access door codes that are associated with user access. We also assign weights to edges based on the spaces to which they are incident.

The state of the system at time t, $State_t(sys)$, is thus defined as the combination of the current location of all users $Loc(u)$ and the state of the environment $State_t(Env)$. The location of the users is defined with respect to the building $\mathbf{B}$, $Loc(U) = \{s | Loc(u) = s \in S, u \in U\}$. The state of the environment $State_t(Env)$ is the condition of the physical and cyber topology and assets (e.g., malfunctioning devices, change in networking access).

4.2 Phase 1: Offline

The framework consists of an offline and an online phase as shown in Figure 2. The offline phase consists of two stages: characterization of users based on their past movement behavior, and construction of models based on users' characteristics and past movement. The inputs to this phase are the past system states $State_{Past} = State_{t_1}(sys) \ldots State_{t_n}(sys)$, and the output of this phase is a collection of tuples $(\mathcal{M}, g)$, in which $\mathcal{M}$ is a model representing the movement behavior of a user and $g : State_t(sys) \rightarrow \{R_i, p_i\}$ is a function that takes in the current system state, and uses the model $\mathcal{M}$ to estimate the probabilities p_i of a user's entering a set of rooms R_i.

User Types The first stage of the offline phase is to distinguish between different users by using their past movement behavior. Typical access control systems assign roles to users based on the sets of rooms that they need to access. However, these roles do not directly reflect the user behavior. Instead, we propose to categorize users according to how they move within a building.

We define the different types of user behavior $\mathbb{T}$ based on the users' "reasons" for movement, where "reason" refers to the context that facilitates users' movement patterns. For each reason or user type $q_i \in \mathbb{T}$, we define a metric λ_i that characterizes the type of behavior that falls under that reason. The metric λ_i takes as input the historical system state pertaining to the user $State_{Past}$ and

[3] This implies that if $e(v_1, v_2)$ exists, the backward edge $e(v_2, v_1)$ also exists in G.

outputs a real number in $\mathbb{R}$. So we map users to user types, $type(u) = q_i \in \mathbb{T}$, by calculating a predicate function on the output of λ_i, $q_i \equiv pred(\lambda_i)$.

Application: In our railway station case study, there are two main types of user movement behavior. The first type, $q_1 \in \mathbb{T}$, involves users who have a very regular movement behavior, of which the primary members are station operators. Station operators work a fixed set of hours in the station, and, because of their job scope, their movement patterns are fairly consistent. They remain in the *Passenger Service Center* (PSC) to assist the public and monitor the state of the station, visit storerooms and staff rooms, and clock in and out.

The second type of users, $q_2 \in \mathbb{T}$, involves those whose movement is triggered when an event occurs. This applies to maintenance staff who visit rooms to conduct maintenance of the equipment. Different maintenance staff members are in charge of different subsystems (e.g., power supply or signaling), and thus they access different sets of rooms in the station.[4]

In order to categorize a user into q_1 or q_2, we define a metric λ_e based on the approximate entropy of a time series [22] constructed using the collected historical access data. The metric is defined as $\lambda_e = ln(C_m/C_{m+1})$, where C_m is the prevalence of repetitive patterns of length m in the time series. Each subsequence of length m in the sequence is compared to other subsequences. If the number of similar subsequences is high, then C_m is large. This metric is shown to be able to quantify the predictability of user movement [23, 24]. We choose m to be 3, which provides a good metric for characterizing our trace as shown in Section 5. If the user's entropy value is low, the user belongs to q_1; otherwise, the user belongs to q_2. In other words, $q_1 \equiv (\lambda_e(u) < \mathbf{E})$, where $\mathbf{E}$ is a numerical threshold. The choice of parameter $\mathbf{E}$ is discussed in Section 5.

User Behavior Models Next, we construct behavior models for each behavior type $q \in \mathbb{T}$ defined earlier. Since each user is motivated to move within the building for different reasons, it is not possible to specify a single model for all users' behavior. Such a model would be inherently biased towards a certain set of users and perform badly for others.

Instead, for each $q \in \mathbb{T}$, we select an appropriate modeling technique $\mathcal{M} \in \mathbb{U}_{\mathcal{M}}$ from a large set of possible modeling techniques $\mathbb{U}_{\mathcal{M}}$. The model should leverage q's distinct characteristics and provide insight into the likelihood that a user will access a room given the current system state $State_{t_{n+1}}(sys)$.

For each user u that has $type(u) = q \in \mathbb{T}$, we learn the model by analyzing past system states $State_{Past}$ in order to assign probabilities to the rooms in $\mathbf{R}$. Finally, we define the function g that takes the state $State_{t_{n+1}}(sys)$ and use the learned model $\mathcal{M}$ to determine the probabilities associated with the user's entering a set of rooms next. Based on $State_{t_{n+1}}(sys)$, $\mathcal{M}$ will calculate and return a set $\{R_i, p_i\}$, where $R_i \in \mathbf{R}$ is a room in the building and $p_i \in [0, 1]$ is the probability that the user will access R_i next.

[4] This may apply to other systems too. E.g., a security guard doing rotations in a building belongs to q_1, and a technical support staff member who goes to an office when his or her assistance is required belongs to q_2.

Application: Users belonging to q_1 have a low entropy value $\lambda_e(u) < \mathbf{E}$. This implies that their movement patterns are highly predictable and repetitive. Thus, we choose to represent a user's movement behavior with a Markov model[5]. The states in the Markov model are the set of rooms $\mathbf{R}$, and a transition from state i to state j implies that a user visits room R_j after R_i.

Given the previous system states $State_{Past}$, we learn the Markov model of a user u. The system state at any point of time $State_{t_i}(sys)$ contains the physical access records for u. We can reconstruct the full movement sequence $Seq(u) = R_1 \dots R_n$ as the sequence of rooms that were visited. The sequence $Seq(u)$ can be divided into segments based on the lengths of the time intervals between consecutive physical accesses. Each segment represents a series of movements that occur close together in time. A period of inactivity (more than 3 hours) separates any two segments. The initial probability vector $\pi(0)$ is the normalized frequency with which each room $r \in \mathbf{R}$ appears at the beginning of each segment of $Seq(u)$. The transition probability p_{ij} is the normalized frequency with which the user visits R_i and then R_j.

However, the users belonging to q_2 have less regular movements and may change movement patterns based on events in the system. So we combine the Markov model with additional contextual knowledge about the states of the devices in the rooms. After vetting the accesses through the Markov model, we correlate the remaining suspicious accesses with logs about device state. Intuitively, if a device in room R_d fails and then a physical access into R_d is logged, that physical access is considered non-malicious. Then, given the device failure incidents in $State_{t_{n+1}}(sys)$, the probability p_d associated with device failures in room R_d in the set $\{R_i, p_i\}$ is suitably changed such that any accesses leading to R_d are considered non-malicious.

4.3 Phase 2: Online

The online phase involves determining, based on the behavior models derived from the offline phase, whether a user's access is an abnormality. The inputs to this phase are the tuples $(\mathcal{M}, g)$ from the offline phase and the current state of the system $State_{t_{n+1}}(sys)$. The output of this phase is a real number in $\mathbb{R}$ that indicates the degree of abnormality of the access. The algorithm for this phase is given below in the OnlineDetection function.

The current system state $State_{t_{n+1}}(sys)$ includes the location of the user $Loc(u) = R_1 \in \mathbf{R}$ and the physical access that is being made, $A = S_1 \rightarrow S_2$, $S_1, S_2 \in S$. In other words, the user is moving from S_1 to S_2. We update the user's behavior model to reflect the current state of the user in the system by computing $g(State_{t_{n+1}}(sys))$. The output is the set $\{R_i, p_i\}$, where $R_i \in \mathbf{R}$ is a room and $p_i \in [0, 1]$ is the probability that the user will access R_i next. Using knowledge of the building topology $\mathbf{B}$, we determine the likelihood that the

[5] Although the Markov model imposes certain assumptions about the movement behavior, such as the memoryless property, it can be extended to include temporal and spatial correlations. We intend to explore these extensions in future work.

access A is anomalous based on the paths from the user's current location to the set of rooms R_i.

Given the access A, we want to determine all the rooms that the user is likely to access. We first find all the rooms that are reachable from S_2, i.e., $P_T = \{R_i | \exists path(S_2, R_i)\}$. For all such vertices $R_i \in P_T$, we decide whether the user is likely to access R_i by moving to S_2 from S_1. If it's easier to access R_i from S_2, then we consider R_i as one of the likely rooms. To decide whether R_i is easily accessed, we calculate path lengths using the weighted edges. We calculate the shortest path from S_1 to R_i, $d(S_1, R_i)$, and compare it to the shortest path from S_1 to R_i through S_2, $d(S_1, S_2) + d(S_2, R_i)$. If the shortest path through S_2 is similar in length to the shortest path, then we consider R_i as a possible room that the user wants to access. With the resulting shortlisted set of rooms, we sum up their likelihoods $\sum_{R_i \in P_T} p_i$ to obtain a final score. If the score is below a threshold value Z, access A is deemed anomalous.

Algorithm 1 OnlineDetection algorithm

```
Require: (Loc(u), A = S1 → S2) ∈ State
  function OnlineDetection(State, g)
    score ← 0; {Ri, pi} ← g(State)
    for all Ri ∈ {Ri, pi} do
      shortestlen ← GetShortestPath(S1 → Ri)
      len ← GetShortestPath(S2 → Ri) + ew(S1, S2)
      if len < shortestlen × k then score ← score + pi end if
    end for
    return score
  end function
```

Application: We keep track of the system state, which is the room that the user has last accessed: $State_{t_{n+1}}(sys) = r_C \in \mathbf{R}$. The function g takes $State_{t_{n+1}}(sys)$ and returns the set of probabilities associated with the next visited room $\{R_i, p_i\}$. In OnlineDetection, we rely on edge weights to calculate path lengths. We assign all edges a weight of 1, with the exception of edges that connect different levels of the building (i.e., staircases, elevators, and escalators). We assume that users prefer to take as few staircases as possible. So we assign a weight of 10 to those edges that connect different levels.

We compare the score returned by OnlineDetection with threshold Z. We choose Z based on the probability distribution p_{r_C}. If the probability distribution has a heavy tail, then there are rooms that the user very rarely visits and may be deemed suspicious. So we choose the threshold value as the 95th percentile. Otherwise, we choose the threshold value as the minimum probability in the distribution. The percentile value can be changed by practitioners based on the system requirements; a higher value reduces the false positives, but potentially malicious movements are missed, while a lower value catches more malicious movements but increases the false positives. Since our results focus more on the trends, the exact value of this threshold is not critical.

5 Evaluation

In this section, we utilize real-world data traces to demonstrate the effectiveness of our framework in our railway transit station case study. First, by evaluating

our usage of the entropy metric, we answer the question of whether the movement behavior of users can be characterized effectively in a complex system. Second, we determine the detection capability of our proposed behavior models. Finally, we examine the possibility of detecting malicious movement in an online manner.

5.1 Experiment Setup

We use a real-world data set containing physical card accesses to a railway station in a city. The duration of the accesses is from June to October 2016. The station has 62 rooms, with a total of 32,100 accesses made by 314 users. While we focus on one station in this work, the whole railway line consists of 33 stations, 12 of which are interchange stations. We estimate that the average number of accesses per hour over all the stations is approximately 450, whereas the highest number of accesses per hour is around 1,200. This poses a significant challenge if the associated logs need to be examined manually.

The data set contains the following information regarding physical accesses: (1) date and time, (2) door code, (3) user identification, and (4) result of access (success or failure). When the access is a failure, it implies either that the user's card had expired or that the user did not have permission to access the room. Those failed accesses serve as ground truth for known abnormal accesses.

We simulated malicious movement in order to conduct a more thorough assessment of the detection ability. For each user, we injected accesses into the testing data. With a certain small probability, we replaced a legitimate access $A = S_1 \rightarrow S$ with a series of injected accesses. We randomly selected a target room $R_T \in \mathbf{R}$ and calculated the shortest path from S_1 to R_T as $S_1 e_1 S_2 \ldots S_n e_n R_T$. For each edge $e_i, i \in [1, n]$ that has a door code, we added an injected access $A_i = S_i \rightarrow S_{i+1}$.

We split the data set into 80% training and 20% testing subsets and performed 10-fold cross-validation. We conducted the experiments on a Windows 7 Home Premium machine with a 2.7 GHz CPU core and 4 GB of RAM.

5.2 Results

In this subsection, we present the evaluation results for our approach from Section 4 based on the physical card accesses data from the railway station.

Implementation Performance We evaluated the running time of both the offline and online phases. The average running time of the construction of Markov models in the offline phase was 33 ms, whereas the average running time of the OnlineDetection function in the online phase was 1.3 ms. The offline phase can be conducted sporadically during system downtime, whereas the online phase is fast enough to be executed in a real-time manner.

Detection Capability Our approach marked 2,975 out of 32,100 accesses as suspicious. Hence, the practitioner's effort would have been reduced by over 90%. Figure 3 shows the number of physical accesses over the ten testing subsets. For each subset, the left bar represents all the accesses, and the right bar represents

the accesses marked as malicious. We can see that most of the injected accesses and malicious ground truth data are detected as malicious. The numbers of false positives are also low and fairly constant over the ten subsets. On average, our approach gives a false positive rate of 0.08 and a false negative rate of 0.34.

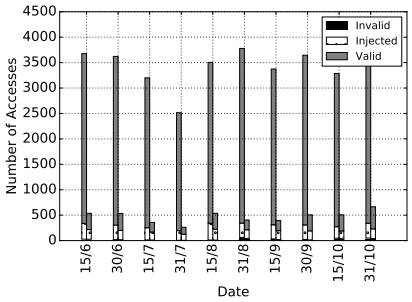


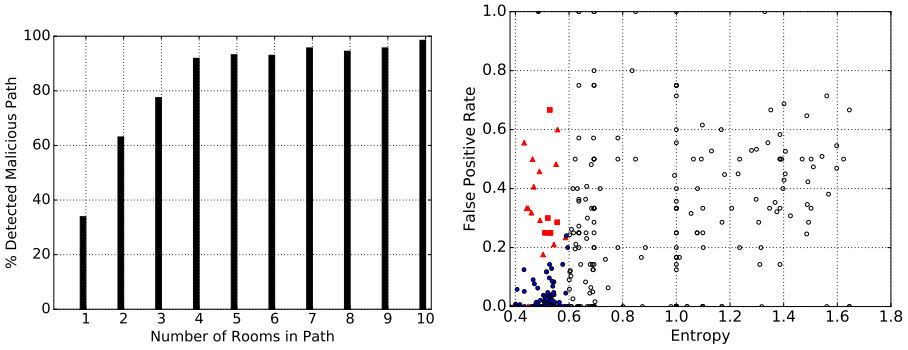
Fig. 3. The number of physical accesses over time. Each tick on the x-axis represents a two-week period; the label indicates the end date of the period. Each bar is divided into three sections representing the valid (or non-malicious) accesses, the malicious ground truth accesses, and injected accesses.

To interpret this result in more detail, we compare our solution with a baseline method that marks any access leading to a previously unvisited room as malicious. It is easy to see that both solutions can identify malicious paths that lead to any previously unvisited room. However, if an attacker carefully selects his or her path by moving only to previously visited rooms, the baseline method will not be able to identify any of those paths (i.e., its false negative rate will be 100%). In comparison, our solution can still raise an alarm if the path covers any unusual transitions among previously visited rooms. The reason that our false negative rate in Figure 3 is relatively high (i.e., 0.34) is that we randomly choose a destination room and generate the shortest path to that destination; thus, a substantial fraction of the generated malicious paths are indistinguishable from legitimate paths that a user actually traveled before. In other words, since most of the generated paths are short, it becomes impossible in a certain fraction of cases to differentiate anomalous and normal movement behavior.

To study how the length of the attacker's path affects the performance of our approach, we experimented with increasing the length of the malicious path we injected (to consider the case when an attacker wanders around the space to do a site survey and explore potential attack opportunities). Instead of injecting paths that ended at a room, we randomly generated paths that went through a sequence of previously visited rooms. We varied the number of visited rooms in the path; the results are presented in Figure 4a. The baseline method is still unable to detect any of these malicious paths, regardless of their lengths. In contrast, the probability of our method's detecting the path approaches 100% as the path grows longer.

User Characterization Next, we evaluate whether the entropy metric defined in Section 4 is suitable for characterizing user behaviors. If the entropy metric can differentiate user behaviors, then the Markov models constructed for users

with low entropy ($q_1 \in \mathbb{T}$) will have a better detection capability (lower false positive and negative rates) than those constructed for users with high entropy. We plot the entropy vs. false positive rates in Figure 4b. Each point in Figure 4b represents a single user in one subset. One user should map to a maximum of 10 points in the plot. We can see that almost all users whose entropy is



(a) (b)

Fig. 4. (a) The percentage of detected malicious paths vs. number of rooms in the path. (b) The distributions of false positive rates with respect to the user's entropy.

below 0.6 (filled markers) have low false positive rates. When the entropy is above 0.6 (unfilled markers), the false positive rates are high. So we can set our entropy threshold **E** to 0.6 in order to distinguish between the user types. Then, 15% of the users would belong to q_1 and 85% to q_2. Although fewer users are categorized under q_1, these users account for 79% of the accesses. Thus, having a low false positive rate for these users implies that entropy is a suitable metric for characterizing user behavior.

However, several outliers show a high false positive rate for q_1-type users. We studied each of them individually and found that there were reasons why the accesses were marked as suspicious. The users represented by triangles in Figure 4b accessed rooms that they had not previously visited, whereas the users represented by squares had a small testing set (< 5 accesses), so their false positive rates are disproportionately high. In actual operation, the training data set and real-time accesses will be much larger, so there won't be outliers.

Integration of Device State In Section 4, we proposed to correlate logs about device state with physical access logs in order to decrease the false positives for the q_2-type users. We assume that the timing information in these logs is synchronized with that in the physical access logs. The logs collected for each type of device (e.g., breaker or lights) are different, and thus the amount of information about the device's state that can be extracted varies. However, we only need to know when a device fails, since the failure could trigger entrance of a maintenance staff member (user of type q_2) into the room to repair the device. In particular, there are four rooms in the station for which we could identify the failure of a device from the logs with particularly strong confidence. These rooms contained devices that controlled the environment in the station (e.g., air

chiller and water pumps). We extracted the textual description and alarm values that indicated device failure and searched the device logs for failure incidents. We compared the timestamps of the failure incidents to the times of the users' accesses. If the user was not in the room prior to the failure, entered the room after the detected failure, and subsequently left the room when the maintenance was complete, then we consider that physical access to be non-malicious. As a result, we reduced the false positives for a subset of the users by an average of 0.45 for the four rooms. This preliminary result shows that we can use additional logs regarding the system environment to determine whether an access is malicious.

Online Detection We determined the feasibility of detecting malicious movement in an online manner by studying how early on a malicious path of a certain length can be detected. We considered injected paths with a length of 4, and the false negative rates for the first, second, third, and fourth injected accesses in the path were 0.54, 0.25, 0.09, and 0, respectively, in our experiment. This shows that our approach is able to detect malicious paths (with a certain minimum length) with high confidence, and even before an attacker reaches the destination room.

6 Discussion and Future Work

In this paper, we present the first step towards understanding how physical access logs can be used to enhance the detection capability of a system. In our case study of railway transit stations, we characterize two different types of user movement behavior. The first user type, q_1, performs well in terms of false positive rates. In ongoing work, we added a notion of time into the states of the Markov model and applied it specifically to the set of station operators in q_1. By separating the station operators into a third user type and honing the model, we have obtained encouraging reductions in false positive rates. For the second user type q_2, the false positive rates are much higher than q_1's. We have shown that using knowledge of the device states improves the false positive rates. However, many issues need to be resolved, such as time drifts between the device logs and physical access logs, differences in contextual understanding of diverse device logs, and missing data regarding device state. We intend to address these issues and pursue this line of thought in future work.

We can also enhance our Markov model further by taking into account the amount of time a user spends in a room, and the function of the room (e.g., storeroom vs. power room). The parameters that we use in our approach can be further tuned and targeted to different users for enhanced detection capability.

In this paper, we only create movement models for each user in isolation. Thus, we do not handle colluding insiders who may tailor their movements such that both parties remain within their movement patterns, but they are able to achieve their malicious goal together. We need to have a more comprehensive view of the system and user movements as a whole in order to tackle such adversaries, and we are currently pursuing this direction by using richer models.

This paper also shows favorable results in using online-based detection in a real-world system. If we can detect a malicious physical access early on in a

user's movement, we can make suitable responses to prevent a potential breach. For example, an administrator can temporarily remove a user's permissions to certain critical rooms, or place the user under further observation.

7 Conclusion

One way in which organizations address insider threats is through physical security. However, the state of the art in building access control is lacking. In this paper, we study the use of physical access logs for detecting malicious movement within a building. We propose a systematic framework that uses knowledge of the system and its users in order to analyze physical access logs. We characterize users by using a set of metrics that take historical physical access data as input. Each user type is mapped to a behavior model, and the details of the model are learned through use of the user's past physical accesses. Finally, we develop an online detection algorithm that takes the behavior model and the building topology as input, and returns a score indicating the likelihood that the user's access is anomalous. We apply our framework to a real-world data trace of physical accesses in railway stations. The results show that our framework is useful in analyzing physical access logs for the purpose of detecting malicious movement.

Acknowledgements. This work was supported in part by the National Research Foundation (NRF), Prime Minister's Office, Singapore, under its National Cybersecurity R&D Programme (Award No. NRF2014NCR-NCR001-31) and administered by the National Cybersecurity R&D Directorate, and supported in part by the research grant for the Human-Centered Cyber-physical Systems Programme at the Advanced Digital Sciences Center from Singapore's Agency for Science, Technology and Research (A*STAR). This work was partly done when Carmen Cheh was a research intern at ADSC. We also want to thank the experts from SMRT Trains LTD for providing us data and domain knowledge.

References

1. Salem, M., Hershkop, S., Stolfo, S.J.: A survey of insider attack detection research. In Stolfo, S.J., Bellovin, S.M., Keromytis, A.D., Hershkop, S., Smith, S.W., Sinclair, S., eds.: Insider Attack and Cyber Security: Beyond the Hacker. Springer (2008) 69–90
2. Alien Vault: Insider threat detection software. https://www.alienvault.com/ (2016)
3. Tripwire: Insider threat security & detection. http://www.tripwire.com/ (2016)
4. CERT Insider Threat Center: Insider threat and physical security of organizations. https://insights.sei.cmu.edu/insider-threat/2011/05/insider-threat-and-physical-security-of-organizations.html (2011)
5. Luallen, M.E.: Managing insiders in utility control environments. Technical report, SANS Institute (2011)
6. Bauer, L., Cranor, L.F., Reeder, R.W., Reiter, M.K., Vaniea, K.: Real life challenges in access-control management. In: Proc. ACM SIGCHI Conference on Human Factors in Computing Systems. (2009) 899–908

7. Kent, A.D., Liebrock, L.M., Neil, J.C.: Authentication graphs: Analyzing user behavior within an enterprise network. Computers & Security **48** (2015) 150–166
8. Pallotta, G., Jousselme, A.L.: Data-driven detection and context-based classification of maritime anomalies. In: Proc. 18th International Conference on Information Fusion. (2015) 1152–1159
9. Radon, A.N., Wang, K., Glasser, U., Wehn, H., Westwell-Roper, A.: Contextual verification for false alarm reduction in maritime anomaly detection. In: Proc. IEEE International Conference on Big Data. (2015) 1123–1133
10. Dash, M., Koo, K.K., Gomes, J.B., Krishnaswamy, S.P., Rugeles, D., Shi-Nash, A.: Next place prediction by understanding mobility patterns. In: Proc. IEEE International Conference on Pervasive Computing and Communication Workshops. (2015) 469–474
11. Gellert, A., Vintan, L.: Person movement prediction using hidden Markov models. Studies in Informatics and Control **15**(1) (2006) 17–30
12. Koehler, C., Banovic, N., Oakley, I., Mankoff, J., Dey, A.K.: Indoor-ALPS: An adaptive indoor location prediction system. In: Proc. ACM International Joint Conference on Pervasive and Ubiquitous Computing. (2014) 171–181
13. Eberle, W., Holder, L.: Anomaly detection in data represented as graphs. Intelligent Data Analysis: An International Journal **11**(6) (2007) 663–689
14. Davis, M., Liu, W., Miller, P., Redpath, G.: Detecting anomalies in graphs with numeric labels. In: Proc. 29th ACM Conf. on Information and Knowledge Management. (2011) 1197–1202
15. Eberle, W., Holder, L., Graves, J.: Detecting employee leaks using badge and network IP traffic. In: IEEE Symposium on Visual Analytics Science and Technology. (October 2009)
16. Liu, C., Xiong, H., Ge, Y., Geng, W., Perkins, M.: A stochastic model for context-aware anomaly detection in indoor location traces. In: Proc. IEEE 12th International Conference on Data Mining. (2012) 449–458
17. Biuk-Aghai, R.P., Si, Y.W., Fong, S., Yan, P.F.: Individual movement behaviour in secure physical environments: Modeling and detection of suspicious activity. In Cao, L., Yu, P.S., eds.: Behavior Computing. Springer (2012) 241–253
18. Hoesl, M.J.: Integrated physical access control and information technology security U.S. Patent No. 6641090 B2, granted on Jun 17 2014.
19. Khurana, H., Guralnik, V., Shanley, R.: System and method for insider threat detection U.S. Patent No. 8793790 B2, granted on Jul 29 2014.
20. Baker, G.: Schoolboy hacks into city's tram system. http://www.telegraph.co.uk/news/worldnews/1575293/Schoolboy-hacks-into-citys-tram-system.html (January 11 2008)
21. Grad, S.: Engineers who hacked into L.A. traffic signal computer, jamming streets, sentenced. http://latimesblogs.latimes.com/lanow/2009/12/engineers-who-hacked-in-la-traffic-signal-computers-jamming-traffic-sentenced.html (December 1 2009)
22. Pincus, S.M.: Approximate entropy as a measure of system complexity. Proceedings of the National Academy of Sciences **88**(6) (1991) 2297–2301
23. Li, X.: Using complexity measures of movement for automatically detecting movement types of unknown GPS trajectories. American Journal of Geographic Information System **3**(2) (2014) 63–74
24. Song, C., Qu, Z., Blumm, N., Barabási, A.L.: Limits of predictability in human mobility. Science **327**(5968) (2010) 1018–1021