Proceedings of the 2017 Winter Simulation Conference
W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds.

MODELING THE EFFECTS OF INSIDER THREATS ON CYBERSECURITY OF COMPLEX SYSTEMS

Teodora Baluta
Lavanya Ramapantulu
Yong Meng Teo
Ee-Chien Chang

Department of Computer Science
National University of Singapore
SINGAPORE

ABSTRACT

With an increasing number of cybersecurity attacks due to insider threats, it is important to identify different attack mechanisms and quantify them to ease threat mitigation. We propose a discrete-event simulation model to study the impact of unintentional insider threats on the overall system security by representing time-varying human behavior using two parameters, *user vulnerability* and *user interactions*. In addition, the proposed approach determines the futuristic impact of such behavior on overall system health. We illustrate the ease of applying the proposed simulation model to explore several ”what-if” analysis for an example enterprise system and derive the following useful insights, (i) user vulnerability has a bigger impact on overall system health compared to user interactions, (ii) the impact of user vulnerability depends on the system topology, and (ii) user interactions increases the overall system vulnerability due to the increase in the number of attack paths via credential leakage.

1 INTRODUCTION

Cybersecurity attacks are growing at an alarming pace (Jang-Jaccard and Nepal 2014, Terada et al. 2016). While there are many technological advances to cope with this challenge, any security system no matter how robust and well-designed ultimately relies on *humans* which become an "Achilles heel" with respect to security (Akhunzada et al. 2015, Guo et al. 2011, Svensson 2013). This overarching issue is due to the complex nature of human behavior within an organizational context (Dhillon and Backhouse 2001, Sasse et al. 2001).

There are many reasons for insider threats or human factor errors that contribute towards security violations (Parsons et al. 2010, Swain and Guttmann 1983). Firstly, employees tend to forget their passwords and hence store them on non-secure places which are easily accessible to others. Secondly, employees fail to distinguish between a compromised or malicious website and disclose their credentials via a phishing link. Thirdly, employees forget to change their passwords after sharing them with their colleague to perform a professional task thus leading to a breach of security or employees share their passwords with their co-workers due to naivety in response to a socially acceptable behavior. For example, a recent survey conducted by Cisco in 2014 indicates that about 25% of employees share passwords with co-workers (CISCO 2014). Hence, it is becoming important to model human behavior and to assess the impact of sharing passwords on system vulnerability.

A key challenge with respect to studying and modeling human behavior is the different perspectives of user interactions. The first challenge is to model interactions with respect to social behaviors with co-workers and the management. Secondly, there is a need to model professional interactions within an organizational context. Thirdly, it is important to model user actions based on both social and professional interactions.

Although there are many aspects of modeling human behavior with respect to user interactions, we address this challenge by using an abstraction of all these variegated interactions modeled as discrete-time stochastic processes.

There are many studies that analyze human interactions for mitigating security threats and focus on modeling human behavior with social and organizational perspectives (Luo et al. 2013, Mouton et al. 2014, Shillair et al. 2015). While these studies provide insights on insider threats and attacks, they do not study the impact of such threats on system vulnerability. On the other hand, many works use probabilistic risk assessment models to derive system vulnerabilities (Holm et al. 2015, Phillips and Swiler 1998). While all of these works either focus on modeling human interactions or modeling system vulnerabilities, to the best of our knowledge, we are the first to model both insider threat and system vulnerability. In contrast to existing works, this paper proposes a discrete-event simulation model, SecureInT (Securing Insider Threats), to study insider threats by modeling both user vulnerabilities as a stochastic process and leakage due to user interactions, and to determine their effects on overall system health or vulnerability. As the proposed model quantifies the degree of human interactions on system vulnerability, it can be used to develop threat mitigation strategies by performing "what-if" analysis to determine effective solutions to combat security threats.

To assess cybersecurity risks, this paper proposes an approach to model unintentional insider threats using two parameters, *user vulnerabilities* and *security leakage* due to user interactions. We illustrate the application of the proposed model by studying an example enterprise system and addressing these research questions: (i) what is the impact of multi-user vulnerability on the derived vulnerability of the system? (ii) what is the impact of leakage via user interactions on the derived vulnerability of the compute system? Modeling the attack progression addresses these questions and provides useful insights on attack paths and how threats can be mitigated by determining defense strategies with the highest impact of minimizing cyberintrusions. In summary, the main contributions of this paper are:

- an approach, SecureInT, to model unintentional insider threats considering both human vulnerabilities and interactions, to represent security leakage and a discrete-event simulation model based on attack-computer and attack-user events to represent potential attack paths
- a discrete-event model with look-ahead analysis to determine the futuristic system health using attack progressions from a given simulation snapshot
- a case study to illustrate the application of the model to draw useful insights on weak links as potential cybersecurity risks in the enterprise system and to determine strategies for mitigating these risks to minimize overall system vulnerability.

The rest of the paper is organized as follows. We present the discrete-event simulation model in Section 2. We apply our model to an example use case in Section 3. We discuss related research works in Section 4 and summarize in Section 5.

2 PROPOSED SECUREINT SIMULATION MODEL

SecureInT (Securing Insider Threats) is a discrete-event simulation model consisting of users, U, an enterprise computer system including its software components, C, and attackers located in the Internet, I, as shown in Figure 1. In contrast to other research works that focus either on modeling human interactions or vulnerabilities (Holm et al. 2015, Luo et al. 2013, Mouton et al. 2014, Shillair et al. 2015), SecureInT models the cybersecurity risk due to unintentional insider threats using two parameters: *user vulnerability*, $v_u(t)$, and *leakage* among users, l_k. An attacker exploits these insider threats with an attack rate of α. Given an attack-event at a given simulation time t, the success factor of this event depends not only on the state of the user at the given time as described in our user behavior model but also on the vulnerability of computer entity, v_c. A successful attack results in a compromised computer entity if the attacker can exploit the user and the computer or directly the computer entity. At any given time t, the output from the simulation is a set of compromised computer entities. While this output is the effect of all the events that

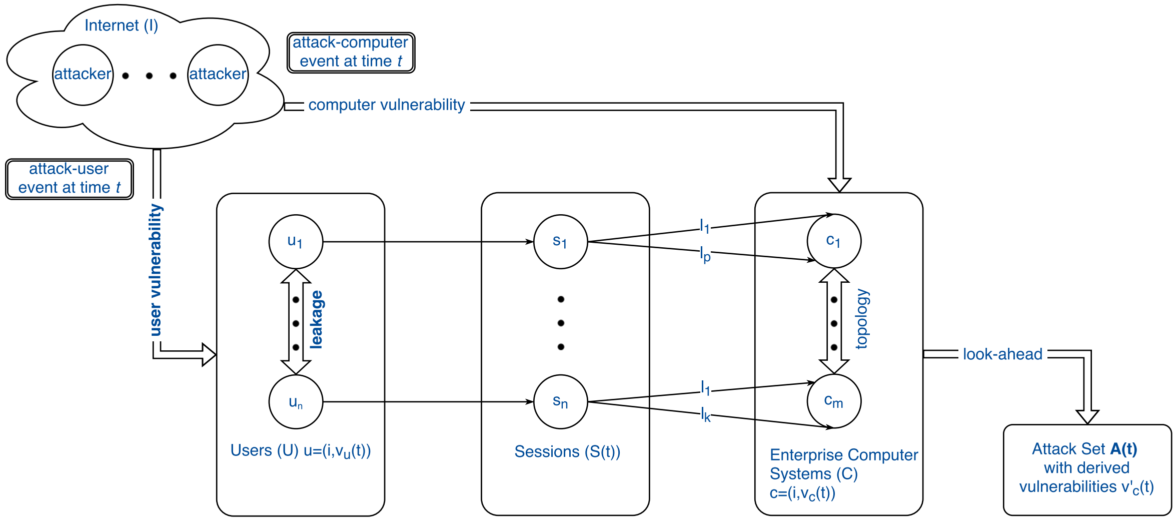
Internet (I)
attacker
attacker
attack-computer event at time *t*
computer vulnerability
attack-user event at time *t*
user vulnerability
u1
leakage
un
Users (U) u=(i,vu(t))
s1
sn
Sessions (S(t))
l1
lp
l1
lk
c1
topology
cm
Enterprise Computer Systems (C) c=(i,vc(t))
look-ahead
Attack Set A(t) with derived vulnerabilities v'c(t)

Figure 1: Overview of SecureInT

have happened till t, we propose to exploit look-ahead analysis to determine the futuristic system health beyond t. The simulation supports multi-stage attacks, i.e. the attacker uses a compromised computer entity within the enterprise system to attack another computer entity that is not directly connected to the Internet. A summary of the notations used in SecureInT is shown in Table 1. We first describe the user entity and its behavior model, followed by the computer entity and the enterprise topology. Next, we discuss in detail the conditions for the attacker to exploit both user and computer entities and lastly, the look-ahead technique.

2.1 User Entity and Behavior Model

In this section, we define the user entity and describe the user behavior model. The user behavior model comprises of security credential leakage due to interactions among users and the different user state transitions during the simulation. We identify two key user behavior parameters: user vulnerability and leakage. The objective of this paper is to demonstrate the usefulness of these two user parameters. Detail modeling of these two parameters is beyond the scope of this paper. The set of users in SecureInT, U, consists of humans who use the computer entities of the enterprise. Each user has a unique identifier and the user entity is defined as:

Definition 1 User A user entity is a tuple, $u = (u_i, v_u(t))$, where $i \in \mathbb{N}$ denotes the user identification number and $v_u(t)$ denotes the user vulnerability probability. The set of users in the enterprise system is $U = \{(u_i, v_u(t)) \mid i \in \mathbb{N}, v_u(t) \in \mathbb{Q} \; and \; v_u(t) \in [0,1)\}$.

The user vulnerability probability is intrinsic to the user and depends on the cognitive limitations based on various psychological aspects. While there are several psychological aspects that lead to social engineering susceptibility, as a proof of concept in this paper, we detail lack of attention, lack of awareness and personality (Greitzer et al. 2014). Lack of attention could be due to overloading of tasks or workplace stress and is changes with time. Lack of awareness involves clicking fake pop-ups and is related to a lack of knowledge about the security threats posed by social engineering attacks. Personality refers to traits such as risk-tolerance, poor risk perception and even personality disorders (Shechter and Lang 2011). We use a combined user vulnerability that varies over time as : $v_u(t) = lack_{attention}(t) + lack_{awareness}(t) + personality$. We consider non-vulnerable users (the ideal case) with $v_u = 0$ but do not consider completely vulnerable users, $v_u < 1$. On the other hand, social interactions among users in an enterprise lead to security breaches such as credential leakage which is discussed next.

Table 1: Model Notations

Symbol	**Description**
Users	
U	set of user entities, u
u	user entity $u = (u_i, v_u)$
$v_u(t)$	user vulnerability at time t
$u_i \xrightarrow{l_k} u_j$	leakage from user u_i to user u_j
λ	user-arrival rate
θ_i	probability of user u_i doing only authorized accesses
Computers	
C	set of computer system entities, c
c	computer system, $c = (c_i, v_c)$
v_c	computer system vulnerability
$c_i \xrightarrow{prot,port} c_j$	network connectivity from client c_i to server c_j
$S(t)$	set of active user sessions, s_i at time t
$s_i(t)$	set of computers with active user session for user u_i at time t
$\sigma_{i,j}$	probability of user u_i having an active session on computer c_j
Attacker	
I	Internet
α	attack rate
$A(t)$	set of compromised systems (attack set) at time t
$I \xrightarrow{prot,port} c_j$	network connectivity from client in Internet to server c_j
$c_j \xrightarrow{prot,port} I$	network connectivity from client c_j to server Internet
P	derived potential attack set using look-ahead
v'_c	derived vulnerability using look-ahead
v_s	overall system health or vulnerability using look-ahead

2.1.1 User-user Interactions: Leakage

To model human interactions due to both social and organizational contexts, we use interactions among users. While interactions between user entities in an enterprise can be derived from an organizational chart or from a social graph, we focus on modeling insider threats such as security leakage due to credential sharing. An example of user-user interaction is that of user u_i sharing credentials with another user u_j. Thus, there is a leakage probability, l_m from $u_i \xrightarrow{l_m} u_j$, such that u_j gains access to the computer entities that u_i has access to. Thus, human interactions in an enterprise network are input to our model to generate the set of accessible set of computer entities per user. This set includes both types of access, directly through authenticated access or indirectly via security leakage because of human interactions. The user-user interaction is a transitive relation, wherein $u_i \xrightarrow{l_m} u_j$ and $u_j \xrightarrow{l_n} u_k$ implies, $u_i \xrightarrow{l_p} u_k$. The leakage probability l_p due to transitivity is derived as, $l_p = l_m * l_n$.

2.1.2 User States

Figure 2 shows the user state transition model with user arrival rate of λ. When a user arrives, it performs authorized accesses with probability θ, and unauthorized access via credential leakage with probability $1 - \theta$. Thus, θ limits the set of computer entities a user conducts an active session via a user session event. A user session event triggers a login of the user to a computer entity and is associated with the pair $(username, password)$. A user u_i can have active sessions with multiple computer entities at a given simulation time instance t.

Definition 2 User Session A user session for user u_i at time t is a set, $s_i(t) = \{c_1, ..., c_n\}$, where $i \in \mathbb{N}$ denotes the user identifier and the user is having an active login session with the collection of computer entities c. This collection c is based on whether the user u_i is authorized or unauthorized depending on θ. The set of user sessions at time t in the system is $S(t) = \{s_i(t) \mid i \in \mathbb{N}\}$.

A user is susceptible to attacks in an active session if the computer entity c of the user login session is connected to either the Internet or to another compromised computer. A successful exploit of the user is based on the attack rate, α, and the user vulnerability v_u at time t of the simulation. Irrespective of whether the exploit is successful, the user transitions to the inactive state with probability σ or remains active in a session with probability $1-\sigma$.

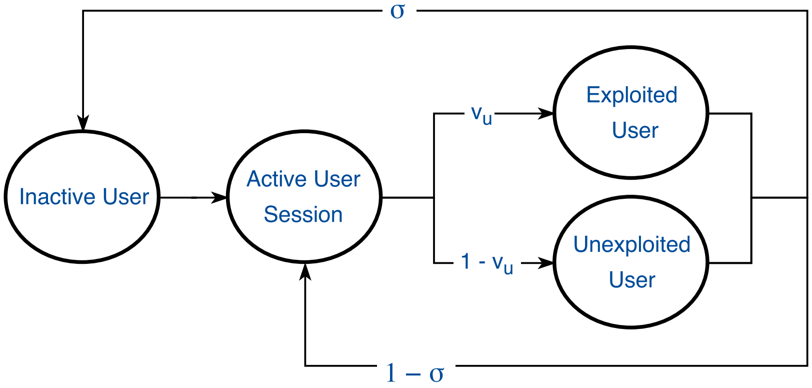

Figure 2: User State-Transition Model

2.2 Enterprise Computer System

Users interact with an enterprise computer system that comprises of computer entities and their interactions constitute the enterprise system's topology. A computer entity is defined as:

Definition 3 Computer Entity A computer entity is a tuple, $c = (c_i, v_c)$ where $i \in \mathbb{N}$ denotes the identifier of the computer entity and v_c denotes the computer's vulnerability probability based on the CVSS score. The set of computer entities in an enterprise system is denoted as the set $C = \{(c_i, v_c) \mid i \in \mathbb{N}, v_c \in \mathbb{Q} \;\; and \;\; v_c \in [0,1)\}$.

The computer's vulnerability v_c is an input to our model and quantifies the probability of a successful attack on a given computer due to existing known vulnerabilities, including hardware or software that can be exploited by an attacker to gain access and/or execute code on the computer. When there is no known vulnerability for a certain computer, the intrinsic vulnerability probability is 0 and we assume that the vulnerability of a computer entity is always less than one.

2.2.1 Computer Entity Interactions: Topology

The topology of an enterprise computer system is imperative in understanding the security risks of a system because more often than not, a multi-stage attack is used for a successful exploit of a computer system vulnerability. While there could be multiple connections among computer entities, from the perspective of modeling insider threats it is important to consider the firewall rules and policies as these could prevent an attack.

Definition 4 Computer Interaction A computer interaction defines a connection between client c_i and server c_j using a protocol (prot) via a port, denoted as $c_i \xrightarrow{prot,port} c_j$ and implies that c_i can initiate a connection defined by the protocol and port.

For example, a well-known service and its respective port is Microsoft's Directory Service running on TCP port 445, which opens a backdoor for remote attackers. If the server has the vulnerability CVE-2008-4250 (cve 2008), then it could be exploited via remote code execution, posing a great risk to the overall system security.

2.3 Attacker

In this section, we first define the attacker followed by the events triggered by the attacker that have the potential to exploit vulnerabilities of computer and user entities to compromise the system.

Definition 5 Attacker An attacker is an entity whose goal is to compromise as many computers inside the enterprise system as possible by exploiting user vulnerabilities and leakage.

For simplicity, we assume that the attacker is located in the Internet, *I*, which is outside of the enterprise system. While there could be multiple attackers, in this model we consider a single attack happening at a given time during the simulation. The model supports an attack rate, α, e.g. the rate of sending phishing links either from the Internet or from a compromised entity within the enterprise. Thus, the SecureInT model supports multi-stage attacks with the attacker first compromising an enterprise computer system, c_i, followed by compromising another enterprise computer system c_j, using the network connectivity from c_i to c_j. The output of the simulation at any given time t is the set of compromised computer entities till time t, denoted by $A(t)$.

Definition 6 Attack Set The attack set consists of all possible computer entities compromised by the attacker till time t of the simulation, $A(t) = \{c_i \mid c_i \in C \land v_c > 0\}$, where c_i is a compromised computer entity.

Next, we discuss the conditions for a successful attack to happen causing a computer entity to be added to the attack set.

2.3.1 Attack-Computer Event

Definition 7 Attack-Computer Event If an attacker has direct access to a computer that has a known vulnerability, the attacker can possibly exploit and gain control over the computer entity at time t when the following conditions are met:

$$(\exists c_i \in C \ \mid \ v_c > 0) \land ((I \xrightarrow{prot,port} c_i) \lor (c_j \in A(t) \ \mid c_j \xrightarrow{prot,port} c_i))$$

While the above conditions are necessary for a successful attack, they are not sufficient. In addition to the above, a successful exploit at a given simulation time t, depends on the probability of an attack at that time which is derived from the attack rate α. At the start of the simulation, we assume that none of the computer entities are compromised. Thus the necessary condition for the very first successful attack is a connection between the attacker in the Internet that can initiate a client request to the exploitable computer entity c_i, $I \xrightarrow{prot,port} c_i$. This is a single stage attack. A multi-stage attack occurs when the computer entity, c_i is compromised via an already compromised enterprise computer c_j, $c_j \xrightarrow{prot,port} c_i$.

2.3.2 Attack-User Event

While the attack-computer event can happen on computer entities that are not part of the user active session, the attack-user event occurs due to unintentional insider threats, for example, user clicking on a phishing link.

Definition 8 Attack-User Event If there is a user u_i that has access to a vulnerable enterprise computer system c_m at time t, then the attack exploits the vulnerability on computer c_m depending on the user u_i's vulnerability v_u at time t, as per the following conditions:

$$\Big(\exists u_i \in U \land v_u(t) > 0 \mid (\exists s_i(t) \in S(t) \ \mid \ \big(c_m \in s_i(t) \ \land c_m \in C \land v_c > 0)\big)\Big) \land$$
$$\Big((c_m \xrightarrow{prot,port} I) \ \lor \big(c_j \in A(t) \ \mid c_m \xrightarrow{prot,port} c_j)\Big)$$

This results in a new enterprise workstation computer, c_m, being added to the compromised computer set, A(t). Similar to the attack-computer event, in addition to the above conditions, a successful exploit at given time t, depends on the attack rate α and the user vulnerability probability v_u at the same time instant t.

2.4 Look-ahead Analysis

While the output from the simulation gives the current system state with the compromised computer entities at a time t, A(t), it is important to determine all potential attacks from this simulation state to mitigate them in a timely manner. The proposed *look-ahead* analysis addresses this by using the entity configurations at time t, and computes all possible attacks to determine the overall system health from the perspective of potential cyberintrusions. The underlying representation for the look-ahead is an attack graph with three types of nodes: *facts* which describe the entities and their interactions, *rule nodes* which represent an attack given a set of preconditions and *privilege nodes* which are the potential compromised computer entities. In contrast to a discrete-event simulation, using this look-ahead as stand-alone technique does not give information on the temporal behavior aspects of users. However, the proposed SecureInT hybrid approach, simulation model with look-ahead, provides a better assessment of the overall system security health to mitigate current and future cybersecurity risks. The output from the look-ahead analysis is the potential attack set defined below.

2.4.1 Potential Attack Set

Definition 9 Potential Attack Set This attack set is output from the look-ahead and consists of all possible computer entities compromised by potential attack paths, $P = \{(c_i, v'_c) \mid c_i \in C \wedge v'_c \in (0,1)\}$, where c_i is a compromised computer entity with a derived vulnerability probability of v'_c.

The overall system health v_s, is derived when look-ahead is invoked from the average of individual vulnerabilities v'_c of each computer entity in the enterprise system. There are risk assessment algorithms (Ou and Singhal 2012) to determine v'_c that quantifies potential cybersecurity risk of an enterprise.

3 MODEL APPLICATION

In this section, we apply our model to a simple enterprise system to gain useful insights on insider threats. First, we describe the system setup followed by a detailed analysis of (i) impact of user vulnerabilities without leakage, (ii) impact of user leakage and (iii) threat mitigation.

3.1 Setup

Figure 3 shows a simplified enterprise system example with two users, Alice and Bob, having access to two workstations. As a measure of security there is a firewall between the enterprise network and the Internet and another firewall restricting access to the database server. Both users are allowed unrestricted outbound traffic, so they can access the Internet and also the organization's web server. The web server (webserver) is directly accessible from the Internet through the HTTP protocol on port 80, while the database server (dbserver) is only accessible from the web server and the two user workstations through MySQL protocol on port 3306. In SecureInT, user entities are represented by set $U = \{(u_1, v_{u_1}(t)), (u_2, v_{u_2}(t)\}$, where u_1 is Alice having at the start of the simulation $v_{u_1}(t = t_{start}) = 0.2$ probability of browsing a malicious or compromised website and u_2 is Bob with the initial vulnerability probability of $v_{u_2}(t = t_{start}) = 0.6$. The computer entity set is $C = \{(c_1, 0.2), (c_2, 0.6), (c_3, 0.9), (c_4, 0.0)\}$. The interactions among computer entities as defined in section Section 2.2.1 are represented in Table 2. In our experiments, three user vulnerability values are chosen, namely, low (l), medium (m) and high (h) that correspond with 0.2, 0.6 and 0.9 respectively.

3.2 Impact of User Vulnerability

In the first analysis, we study the impact of user vulnerability alone without leakage. We assume Alice and Bob are vulnerable users but do not share confidential information like credentials between them. We first increase the vulnerability of Bob (v_{u_2}), keeping the vulnerability of Alice (v_{u_1}), fixed during the first part of the simulation. During the second simulation phase, we increase v_{u_1} to m and increase v_{u_2} from l to h and repeat this with $v_{u_1} = h$ during the last simulation phase. Intuitively, it is expected that an increase in Bob's user vulnerability will result in an increase in system vulnerability. However, as shown in Table 3, the *average system vulnerability* (v_s) does not vary with Bob's vulnerability (v_{u_2}) but only depends on Alice's vulnerability (v_{u_1}). The web server's derived vulnerability remains the same because the attack event for

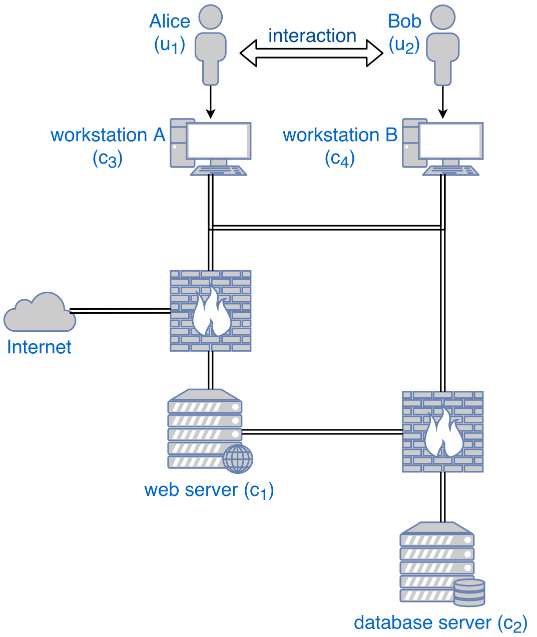

Figure 3: A Simple Enterprise System (Ou et al. 2005), (Ou and Singhal 2012)

Table 2: Entity Interactions

Entities	Interactions
webserver (c_1)	$I \rightarrow c_1$ $c_1 \rightarrow c_2$
dbserver (c_2)	-
workstation A (c_3)	$c_3 \rightarrow I$ $c_3 \rightarrow c_1$ $c_3 \rightarrow c_2$
workstation B (c_4)	$c_4 \rightarrow I$ $c_4 \rightarrow c_1$ $c_4 \rightarrow c_2$

Table 3: User Vulnerability with No Leakage

Users Vulnerability Probabilities		Derived Vulnerability Probabilities			
v_{u_1}	v_{u_2}	v_s	v_{c_2}	v_{c_1}	v_{c_3}
l	l	0.21	0.22	0.20	0.21
l	m	0.21	0.22	0.20	0.21
l	h	0.21	0.22	0.20	0.21
m	l	0.39	0.40	0.20	0.58
m	m	0.39	0.40	0.20	0.58
m	h	0.39	0.40	0.20	0.58
h	l	0.52	0.52	0.20	0.83
h	m	0.52	0.52	0.20	0.83
h	h	0.52	0.52	0.20	0.83

the webserver is an attack-computer event and users do not influence this event. Bob's vulnerability (v_{u_2}) does not influence the derived vulnerabilities of workstation A and the dbserver because his session uses workstation B that does not expose any known vulnerability.

However, as workstation A is accessed by Alice who becomes more and more vulnerable, we see the impact in our results. As Alice's vulnerability (v_{u_1}) increases from low to high, the overall system vulnerability (v_s) increases by 2.5 times. Thus, the proposed simulation model can be applied to study insider threats with respect to the sphere of influence that the insider has in the network.

Observation 1. Increasing the vulnerability of users does not impact system vulnerability when users access computers that do not expose any vulnerability and the users do not interact with each other.

3.3 Impact of User Leakage

Next, we study the impact of leakage when users are vulnerable. We assume Alice and Bob, besides being individually vulnerable, are also prone to share credentials between them with low probability, ($u_1 \xrightarrow{l_{1,2}} u_2$) with $l_{1,2} = 0.2$. Figures 4 and 5 show the impact of user vulnerabilities with and without leakage for the database server and workstation A, respectively using simulation time snapshots. For both, the database server and workstation A, the derived vulnerabilities increase (up to 30% for the database server and up to 70% for the workstation) because leakage allows for more number of possible attack-user events and thus increasing the system vulnerabilities.

With respect to the impact, similar to results in Section 3.2, user u_1 has a much bigger impact than user u_2. While increasing u_2's vulnerability from low to high results in a 16% increase in the database server's vulnerability, increasing u_1's vulnerability from low to high causes the database server's vulnerability to increase by two times. Similarly, for workstation A, u_1 has a higher impact because it causes an increase by two times, while u_2 affects the system vulnerability by only 40%.

Observation 2. While users with access to non-vulnerable computers do not impact system vulnerability, leakage of security credentials to such users does increase system vulnerability. However, vulnerability of users with direct access impacts the system vulnerability more than the vulnerability of users who gain access via leakage.

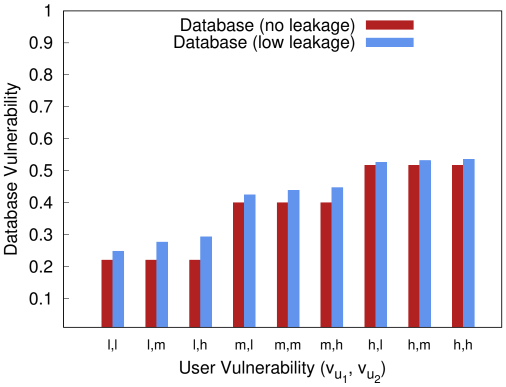

Figure 4: Increase of Security Risk of the Database Server with User Leakage

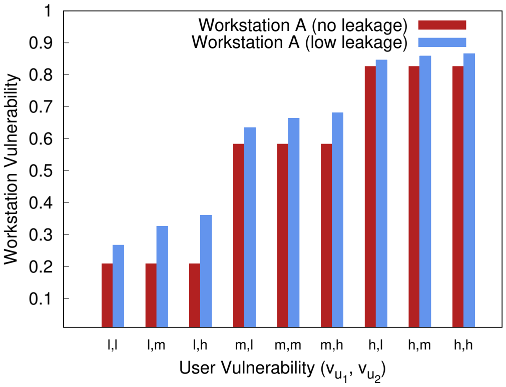

Figure 5: Increase of Security Risk of the Workstation with User Leakage

3.4 Threat Mitigation

In this section, we first consider a defense strategy adopted to mitigate insider threats with user vulnerability alone and without leakage. Section 3.2 showed that there is considerable increase of risk in this case with respect to workstation A and the database server. Here we consider options to mitigate the increased risk. While risk mitigation with respect to security hardening of computer systems has been studied (Ou and Singhal 2012), we consider security hardening of users. The results from Section 3.2 show that user u_2 does not impact system vulnerability, hence a strategy involving security hardening by training u_2 is not useful. Thus, we can apply our model to identify the right user for security training, not only decreasing risk but also saving training costs.

Secondly, we consider the threat mitigation strategy for the case with user vulnerability and leakage from u_1 to u_2. Section 3.3 shows that leakage impacts system vulnerability and thus, it is important to consider security training for both users u_1 and u_2. However, with respect to security training cost, it is useful to determine training of which user will have a bigger impact in reducing risk. The results from Section 3.3 indicate that the impact of user u_1 is significantly higher compared to u_2 and thus we can use these results to choose whom to train. Additionally, training u_1 reduces the risk of leakage from u_1 to u_2. *Observation 3. Users are not equal when it comes to impact on system vulnerabilities. User's influence depends on the network topology as well as user privileges and access control lists.*

4 RELATED WORK

While the current state-of-the-art either model system vulnerabilities or user interactions, we model both and evaluate the impact user interactions on system vulnerabilities. Most of the modeling approaches with respect to cyber attacks and risk mitigation make use of attack graphs to represent system topologies and to evaluate the impact of vulnerabilities on specific networks (Phillips and Swiler 1998, Ou et al. 2005, Ritchey and Ammann 2000, Sheyner et al. 2002). In contrast, we propose a discrete-event simulation model for attack paths and represent both user-system vulnerabilities and user interactions based on security leakage in addition to determining overall system health using look-ahead. While attack graphs provide static snapshots of system vulnerabilities, SecureInT is designed to address the dynamic change in user and system vulnerability over time.

Although many works focus on quantifying existing vulnerabilities, some research works predict potential system vulnerabilities that are yet to be found. One approach towards vulnerability predictive models is the use of time series aggregations with previously recorded data breach incident reports (Condon et al. 2008, Alhazmi et al. 2007, Box and Jenkins 1990). Other approaches use machine learning algorithms for classification and prediction of vulnerabilities and exploits (Bozorgi et al. 2010, Colbaugh and Glass 2011), mining existing vulnerability databases for coding patterns (Neuhaus et al. 2007), natural language processing techniques to scout for vulnerabilities (Mokhov et al. 2014) and stochastic models based on static software code analysis (Rahimi and Zargham 2013). New vulnerabilities predicted by these works complement our approach and can be used as inputs to the model.

There are many studies that focus on perceived cybersecurity risks due to interactions on social networking sites (Saridakis et al. 2016). While these studies focus on one aspect of user interaction and delve into details of behavioral modeling with respect to these online networks, this paper models user-user interactions which subsumes social networks. Other behavioral aspects among users causing them to be insider threats include, organization-employee trust relationship (Guo et al. 2011, Kirlappos and Sasse 2014), mismorphisms which are failures in preserving the structure when moving from the perceived workflow to the realistic usage model of cyber technologies (Blythe et al. 2013, Smith et al. 2015). These works complement our approach as the results from such studies on human behavior can be used as inputs to the user entity described in Section 2.1.

There are many studies involving user credentials such as passwords, and the fact that users are unable to cope with the increasing number of passwords and their complexity leads to an increase in the number of insider threats (Inglesant and Sasse 2010, Shay et al. 2010, Hayashi and Hong 2011). Many studies involve user behaviors in organizations focusing on password policies and sharing credentials (Adams and Sasse 1999, Kaye 2011, Komanduri et al. 2011). These studies motivate the work in this paper as they provide empirical evidence for the interactions among users with respect to sharing credentials and provide insights into realistic values for the leakage attribute.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a discrete-event simulation model, SecureInT, to investigate the effect of insider threats on system vulnerabilities. While the current state-of-the-art either models user or system vulnerabilities, SecureInT considers both users and computer systems and their interactions. SecureInT model is useful for performing ”what-if” analysis and to gain insights on strategies to mitigate potential cyberintrusions. We applied our model to demonstrate how user vulnerability and leakage combined increase threats, and under what conditions these factors do not increase cybersecurity risks. As an example, we show that leakage of Alice’s credentials to Bob increases the system vulnerability by up to 70%. Thus, the proposed model can be applied to garner what security enhancements (or hardening) will be most optimal to decrease the cumulative risk for a given system topology.

We identified two key user behavior parameters, user vulnerability and user leakage, and through a probability model we demonstrated the usefulness of these two parameters. Further work is required to model these two parameters to reflect the changes in user behavior over time and to validate the model. While this paper shows a simplistic two user case, further work is needed to scale the number of users.

ACKNOWLEDGMENTS

This research is supported by the National Research Foundation, Prime Ministers Office, Singapore under its National Cybersecurity R&D Program (Award No. NRF2015-NCRNCR002-001) and administered by the National Cybersecurity R&D Directorate.

REFERENCES

2008. “CVE 2008-4250”. *online: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2008-4250*.

Adams, A., and M. A. Sasse. 1999, December. “Users Are Not the Enemy”. *Commun. ACM* 42 (12): 40–46.

Akhunzada, A., M. Sookhak, N. B. Anuar, A. Gani, E. Ahmed, M. Shiraz, S. Furnell, A. Hayat, and M. K. Khan. 2015. “Man-At-The-End Attacks: Analysis, Taxonomy, Human Aspects, Motivation and Future Directions”. *Journal of Network and Computer Applications* 48:44–57.

Alhazmi, O. H., Y. K. Malaiya, and I. Ray. 2007, May. “Measuring, Analyzing and Predicting Security Vulnerabilities in Software Systems”. *Comput. Secur.* 26 (3): 219–228.

Blythe, J., R. Koppel, and S. W. Smith. 2013. “Circumvention of Security: Good Users do Bad Things”. *Security & Privacy* 5 (11): 80–83.

Box, G. E. P., and G. Jenkins. 1990. *Time Series Analysis, Forecasting and Control*. Holden-Day, Incorporated.

Bozorgi, M., L. K. Saul, S. Savage, and G. M. Voelker. 2010. "Beyond Heuristics: Learning to Classify Vulnerabilities and Predict Exploits". In *Proc. of the 16th SIGKDD*, 105–114.

CISCO 2014, March. "Data Leakage Worldwide: Common Risks and Mistakes Employees Make". *White Paper*.

Colbaugh, R., and K. Glass. 2011. "Proactive Defense for Evolving Cyber Threats". In *Proc. of Intelligence and Security Informatics Conference*, 125–130.

Condon, E., A. He, and M. Cukier. 2008. "Analysis of Computer Security Incident Data Using Time Series Models". In *Proc. of the 19th International Symposium on Software Reliability Engineering*, ISSRE '08, 77–86. Washington, DC, USA.

Dhillon, G., and J. Backhouse. 2001. "Current Directions in IS Security Research: Towards Socio-organizational Perspectives". *Information Systems Journal* 11 (2): 127–153.

Greitzer, F. L., J. R. Strozer, S. Cohen, A. P. Moore, D. Mundie, and J. Cowley. 2014. "Analysis of unintentional insider threats deriving from social engineering exploits". In *Proc. of SPW*, 236–250.

Guo, K. H., Y. Yuan, N. P. Archer, and C. E. Connelly. 2011. "Understanding Nonmalicious Security Violations in the Workplace: a Composite Behavior Model". *Journal of Management Information Systems* 28 (2): 203–236.

Hayashi, E., and J. Hong. 2011. "A Diary Study of Password Usage in Daily Life". In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, 2627–2630.

Holm, H., K. Shahzad, M. Buschle, and M. Ekstedt. 2015. "P CySeMoL: Predictive, Probabilistic Cyber Security Modeling Language". *Transactions on Dependable and Secure Computing* 12 (6): 626–639.

Inglesant, P. G., and M. A. Sasse. 2010. "The True Cost of Unusable Password Policies: Password Use in the Wild". In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, 383–392.

Jang-Jaccard, J., and S. Nepal. 2014. "A Survey of Emerging Threats in Cybersecurity". *Journal of Computer and System Sciences* 80 (5): 973–993.

Kaye, J. J. 2011. "Self-reported Password Sharing Strategies". In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, 2619–2622.

Kirlappos, I., and M. A. Sasse. 2014. "What Usable Security Really means: Trusting and Engaging Users". In *International Conference on Human Aspects of Information Security, Privacy, and Trust*, 69–78. Springer.

Komanduri, S., R. Shay, P. G. Kelley, M. L. Mazurek, L. Bauer, N. Christin, L. F. Cranor, and S. Egelman. 2011. "Of Passwords and People: Measuring the Effect of Password-composition Policies". In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, 2595–2604.

Luo, X. R., R. Brody, A. Seazzu, and S. Burd. 2013. "Social Engineering: The Neglected Human Factor for Information Security Management". *Managing Information Resources and Technology: Emerging Applications and Theories: Emerging Applications and Theories*:151.

Mokhov, S. A., J. Paquet, and M. Debbabi. 2014. "The use of NLP Techniques in Atatic Code Analysis to Detect Weaknesses and Vulnerabilities". In *Canadian Conference on Artificial Intelligence*, 326–332. Springer.

Mouton, F., L. Leenen, M. M. Malan, and H. Venter. 2014. "Towards an Ontological Model Defining the Social Engineering Domain". In *IFIP International Conference on Human Choice and Computers*, 266–279. Springer.

Neuhaus, S., T. Zimmermann, C. Holler, and A. Zeller. 2007. "Predicting Vulnerable Software Components". In *Proc. of the 14th ACM Conference on Computer and Communications Security*, 529–540.

Ou, X., S. Govindavajhala, and A. W. Appel. 2005. "MulVAL: A Logic-based Network Security Analyzer". In *Proc. of the 14th Conference on USENIX Security Symposium - Volume 14*, 8–8: USENIX Association.

Ou, X., and A. Singhal. 2012. *Quantitative security risk assessment of enterprise networks*. Springer.

Parsons, K., A. McCormac, M. Butavicius, and L. Ferguson. 2010. "Human Factors and Information Security: Individual, Culture and Security Environment". Technical report, DTIC Document.

Phillips, C., and L. P. Swiler. 1998. “A graph-based system for network-vulnerability analysis”. In *Proc. of the workshop on New security paradigms*, 71–79.

Rahimi, S., and M. Zargham. 2013, June. “Vulnerability Scrying Method for Software Vulnerability Discovery Prediction Without a Vulnerability Database”. *Transactions on Reliability* 62 (2): 395–407.

Ritchey, R. W., and P. Ammann. 2000. “Using Model Checking to Analyze Network Vulnerabilities”. In *Proc. of the Symposium on Security and Privacy*, 156–.

Saridakis, G., V. Benson, J.-N. Ezingeard, and H. Tennakoon. 2016. “Individual Information Security, User Behaviour and Cyber Victimisation: An Empirical Study of Social Networking Users”. *Technological Forecasting and Social Change* 102:320–330.

Sasse, M. A., S. Brostoff, and D. Weirich. 2001. “Transforming the ’Weakest Link’-A Human/Computer Interaction Approach to Usable and Effective Security”. *BT technology journal* 19 (3): 122–131.

Shay, R., S. Komanduri, P. G. Kelley, P. G. Leon, M. L. Mazurek, L. Bauer, N. Christin, and L. F. Cranor. 2010. “Encountering Stronger Password Requirements: User Attitudes and Behaviors”. In *Proc. of the Sixth Symposium on Usable Privacy and Security*, 2:1–2:20.

Shechter, O. G., and E. L. Lang. 2011. “Identifying personality disorders that are security risks: Field test results”. Technical report, DTIC Document.

Sheyner, O., J. Haines, S. Jha, R. Lippmann, and J. M. Wing. 2002. “Automated generation and analysis of attack graphs”. In *Proc. of Symposium on Security and Privacy*, 273–284.

Shillair, R., S. R. Cotten, H.-Y. S. Tsai, S. Alhabash, R. LaRose, and N. J. Rifon. 2015. “Online Safety Begins with you and me: Convincing Internet Users to Protect Themselves”. *Computers in Human Behavior* 48:199–207.

Smith, S., R. Koppel, J. Blythe, V. Kothari, and V. H. Kothari. 2015. “Mismorphism: a Semiotic Model of Computer Security Circumvention (Extended Version)”. *Computer Science Technical Report TR* 768.

Svensson, G. 2013. “Auditing the Human Factor as a Part of Setting up an Information Security Management System”.

Swain, A. D., and H. E. Guttmann. 1983. “Handbook of Human-reliability Analysis with Emphasis on Nuclear Power Plant Applications. Final Eeport”. Technical report, Sandia National Labs., Albuquerque, NM (USA).

Terada, T., Y. Katayama, S. Torii, and H. Tsuda. 2016. “Security Measures Based on Human Behavior Characteristics”. *FUJITSU Sci. Tech. J* 52 (3): 78–84.

AUTHOR BIOGRAPHIES

TEODORA BALUTA is a PhD candidate in Computer Science at School of Computing, National University of Singapore (NUS). Her email address is teobaluta@comp.nus.edu.sg.

LAVANYA RAMAPANTULU is a Research Fellow at the National Cybersecurity R&D Lab, Singapore. Her email address is lavanya@comp.nus.edu.sg.

YONG MENG TEO is an Associate Professor with the Department of Computer Science at the National University of Singapore (NUS), and an Affiliate Professor at the NUS Business Analytics Center. He heads the Computer Systems Research Group. His email address is teoym@comp.nus.edu.sg.

EE-CHIEN CHANG is an Associate Professor with the Department of Computer Science at the National Universityof Singapore (NUS). His email address is changec@comp.nus.edu.sg.