

An Experience Using System Dynamics to Facilitate an Insider Threat Workshop

Andrew P. Moore, apm@cert.org, 412-268-5465
Dawn M. Cappelli, dmc@cert.org, 412-268-9136
Hannah Joseph, hjoseph@cmu.edu, 315-560-2528¹
Randall F. Trzeciak, rft@cert.org, 412-268-7040

CERT^{®2}, Software Engineering Institute and
CyLab at Carnegie Mellon University
4555 Fifth Avenue
Pittsburgh, PA 15213

Abstract

CERT has been investigating the use of system dynamics to better understand the threat to an organization's information technology (IT) systems posed by malicious employees or contractors of that organization. At the 2006 International System Dynamics Conference (ISDC 2006) we published a system dynamics model that was originally intended to be used as an interactive simulation in a workshop on insider threat. We believed that the model and simulation would effectively communicate the risks and mitigations involved in the insider threat problem. Through several pilots of the model-based workshop we learned how to better use system dynamics modeling as the basis for communicating complex concepts within the insider threat domain to an audience of business and IT managers who are neither familiar with nor interested in becoming familiar with system dynamics modeling. This paper describes the MERIT³ model as well as the development and evolution of the insider threat workshop based on this model.

1 Introduction

The CERT Program at Carnegie Mellon University's Software Engineering Institute has been investigating the use of system dynamics to better understand and communicate the threat to an organization's information technology (IT) systems posed by malicious current or former employees or contractors. Our work began with a collaborative group modeling workshop on insider threat hosted by CERT and facilitated by members of what has evolved into the Security Dynamics Network and the Special Interest Group on Security of the System Dynamics Society (Anderson et al. 2004). Work since then has involved modeling insider fraud (Rich et al. 2005), insider IT sabotage (Cappelli et al. 2006), and espionage (Band et al. 2006). Prior work had involved modeling a specific insider IT sabotage case (Melara et al 2003).

¹ Hannah Joseph is also a student at the Information Networking Institute, Carnegie Mellon University.

² CERT and CERT Coordination Center are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

³ The CyLab *MERIT* (Management and Education of the Risk of Insider Threat) project is supported by the Army Research Office through grant number DAAD19-02-1-0389 ("Perpetually Available and Secure Information Systems") to Carnegie Mellon University's CyLab.

In addition to valuable insights in these domains gained from our modelling efforts, we have also made interesting discoveries pertaining to the use of system dynamics to facilitate training, particularly in the insider IT sabotage domain. At the 2006 International System Dynamics Conference (ISDC 2006) we published a system dynamics model for insider IT sabotage, a subclass of insider threat that involves the use of information technology to direct specific harm at an organization or individual (Cappelli et al. 2006). The model, which we call *MERIT* (Management and Education of the Risk of Insider Threat), and its simulation was originally intended to support interactive discussions in a workshop on insider threat. We believed that the model and simulation would effectively communicate the risks and mitigations involved in the insider threat problem. Through several pilots of the model-based workshop, we developed other alternatives to better use system dynamics modeling as the basis for communicating complex concepts within the insider threat domain to an audience of business and IT managers who are neither familiar with nor interested in becoming familiar with system dynamics modeling.

This paper describes the *MERIT* insider IT sabotage model as it evolved through a series of group modeling efforts, the evolution of an insider threat workshop based on the model, and several pilots of the workshop. We also describe our approach, within the context of the workshop, to introducing insider threat domain concepts prior to presenting the model to workshop participants. The model is used to bring into sharper focus the concepts previously described, with an emphasis on their dynamic interrelationship. The next sections describe the model of the insider IT sabotage problem and problem mitigations now used in the workshop. We conclude with an assessment of our modelling efforts and a summary of our ongoing and future work in the area. Additionally, appendices describe background in the system dynamics approach, an instructional case used in our insider threat workshop, the insider IT sabotage model, and a glossary of terms used throughout this paper.

2 Insider IT Sabotage Model and Workshop Evolution

This section briefly summarizes the development of the *MERIT* insider IT sabotage model, which is followed by the development and evolution of a workshop based on the model. The CERT team has gathered an extensive library of actual insider threat cases in our work with the U.S. Secret Service and the U.S. Department of Defense⁴. Therefore, one of the requirements for our insider threat models was that they be empirically based (i.e., developed based on actual insider threat cases and validated against those cases). The first step in our modeling effort was to determine the relevant cases for the modeling effort. Although 49 insider IT sabotage cases were examined for the CERT/Secret Service *Insider Threat Study*, not all of them contained the type and depth of information required for this modeling effort. The *Insider Threat Study* focused on obtaining answers to hundreds of discrete questions regarding the psychological and technical aspects of the cases. On the other hand, the information needed for this modeling exercise was somewhat different, involving the dynamic nature of key variables. In the end, 30 insider IT sabotage cases were selected for use in this project based on availability of pertinent information.

Figure 1 depicts the evolution of the system dynamics model. The source of information for the modeling was the set of 30 insider IT sabotage cases (shown in the center of Figure 1). The team

⁴ For the remainder of this report, this study is referred to as the *Insider Threat Study*. See the CERT Insider Threat website for a description of CERT's insider threat research with the U.S. Secret Service and Department of Defense: http://www.cert.org/insider_threat/

captured information relevant to the modeling effort in the *Sabotage Case Details Database* (stage 1). Group modeling efforts (stage 2) took us to a *Detailed Sabotage Concept Model*. The group that developed this initial model included CERT personnel with extensive knowledge of the IT sabotage cases from the *Insider Threat Study*. These team members had the same general understanding of the cases involved. A psychologist and experienced system dynamicist acted as facilitator for the group effort.

We experimented with numerous formulations before identifying factors that conveyed the most important concepts from the sabotage cases we reviewed. A critical point was our presentation in January 2006 to members of the Security Dynamics Network (SDN) and the Security SIG of the System Dynamics Society at the University at Albany in New York. After presenting the detailed concept model, the group suggested simplifying the model by focusing on the key dynamics of the core problem. This focus would enable us to coherently teach the key lessons to workshop participants without getting bogged down by extraneous details. The final product of this effort was a much improved model, namely the *Re-oriented Sabotage Simulation Model* (stage 3). Executions of this model represented the problematic behaviors seen in a majority of the cases selected for study, as confirmed through our validation efforts (stage 4). This model was the one that the group published in the proceedings of ISDC 2006 (Cappelli et al. 2006).

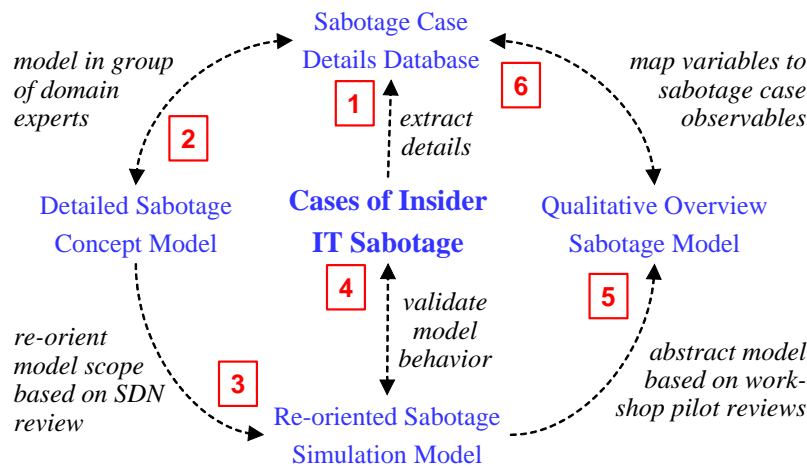


Figure 1: Sabotage Model Evolution

Next, we developed presentation materials for the workshop that encompassed the findings of the *Insider Threat Study*, the model and how it relates to those findings, and the simulation of the model demonstrating how the problematic behavior evolves over time. Throughout the summer and fall of 2006, we conducted a series of workshop pilots involving managers representing information technology, human resources, legal council, risk management, and physical security across a wide range of organizations. The feedback from these pilots emphasized the need to simplify the presentation of the model in the context of the workshop. The model should be used not for introducing domain concepts but for describing the complex relationships between those concepts.

Based upon the feedback, we organized the workshop so that the most critical domain concepts were presented prior to introducing the system dynamics model to the participants. We simplified the notation used to present the model and focused on the feedback relationships between model

variables. The resulting model is the *Qualitative Overview Sabotage Model* (stage 5). We also made sure all model variables were linked to case observables (stage 6). The term *observable* in this report refers to specific events, conditions, or individual and organization actions that could have been observed in the cases examined. The linkage to observables relates behaviors recognized as important for early detection to actions managers can take to better identify and understand an evolving insider threat. This approach helps ensure that recommendations made as a result of the modeling effort are actionable.

Early workshop participants suggested that we mostly restrict the notation to that needed to represent causal loop diagrams. Arrows still represent the influence of one variable on another. However, we found that explicitly signing arrows with an indicator of positive or negative influence caused some confusion. Many participants preferred that the nature of the influences and causal loops themselves (i.e., whether they were balancing or reinforcing) come out in the verbal presentation of the model and not in its graphical depiction. We decided on a middle ground: to use a more subtle indicator of influence type, one that the audience could easily ignore if they choose to do so. For the presentation in this paper, we use *solid* arrows for *positive* influence and *dashed* arrows for *negative* influence. Background on the system dynamics methodology is provided in Appendix A for those unfamiliar with the approach.

3 Workshop Introduction of Domain Concepts

As mentioned in the last section, we found it important to introduce insider threat domain concepts prior to presenting the model to workshop participants. Our insider IT sabotage workshop has the following structure:

- *Insider Threat Study* overview
- Interactive discussion of the instructional case of insider IT sabotage
- General observations from the case data
- System dynamics model: problem, prevention, and mitigation
- Recommendations for countering the threat

We introduce domain concepts during the *Insider Threat Study* overview and in the interactive instructional case discussion. By the time we introduce the model, participants are very familiar with the primitive domain concepts. The model then serves to bring into sharper focus the concepts previously described, with an emphasis on the dynamic interrelationship of those concepts. This approach helps ensure that workshop participants are not overwhelmed with too many new concepts, both modeling and domain, at the same time. The rest of this section provides an overview of our instructional case and general observations about insider IT sabotage. We present this material to illustrate how domain concepts are presented prior to presenting the model.

3.1 The Instructional Case

A concrete case example helps to clearly illustrate the relationship between the aspects of the insider threat and the effectiveness of various measures to counter the threat. However, the sensitivity of actual *Insider Threat Study* case data precludes the use of actual cases for training. We, therefore, developed a fictional case scenario that is representative of a preponderance of actual cases of insider IT sabotage from the *Insider Threat Study*. The fictional organization is

called iAssemble, Inc.⁵ The full text of the iAssemble case example provided in Appendix A is a substantial revision of a previously published version (Cappelli et al. 2006) based on specific guidelines in the area (Naumes and Naumes 1999). Moore et al. (2007) provides guidelines for using the instructional case in a classroom setting.

We believe that the iAssemble case provides a coherent and well-grounded basis for training on the important issues relevant to insider IT sabotage and is representative in character (but not necessarily detail) of many of the actual cases that we have seen. This fictional case deals with the events surrounding an insider IT sabotage case at iAssemble. Ian Archer was hired during the company startup as a computer specialist and technical assistant to the original founders. With hard work and dedication, he became the sole system administrator at iAssemble, a position he held for four years. He was also responsible for building the software that ran the company's computer systems. With the increase in sales at iAssemble and its focus on making profits and meeting deadlines, iAssemble began hiring new people.

After being passed over for the new lead system administrator position, Archer began acting out in the workplace. In the next few months, Archer's disruptions to iAssemble operations grew to the point that iAssemble managers decided they had no choice but to let him go. On the day he was fired, Archer installed a malicious software program, which is generically referred to as a logic bomb, on iAssemble's central server. The logic bomb, which detonated one week after Archer's dismissal, deleted all of the programs supporting iAssemble's mission critical processes that Archer himself had developed. The instructional case describes in more detail the motive behind the attack, the (non)-technical actions taken by the insider and the organization, and the impact of those actions on iAssemble.

The following four questions are used to focus the workshop participant's attention on the issues and concepts central to understanding insider IT sabotage. They focus on

- identifying behavioral and technical precursors exhibited by the insider
- understanding how Archer's personal predispositions and unmet expectations caused an escalation of disgruntlement that was triggered by a precipitating event
- technical discussions regarding the access paths into the organization's systems that are available to the insider.

Question#1: Why did Archer attack iAssemble?

The concepts of unmet expectation and personal predisposition are critical to understanding why Archer attacked iAssemble. These concepts are defined as follows.

- *Unmet expectation*: An unsatisfied assumption by an individual that an organization action or event will (or will not) happen, or a condition will (or will not) exist
- *Personal predisposition*: a characteristic historically linked to a propensity to exhibit malicious insider behavior⁶.

The root cause of Archer's disgruntlement was his unmet expectation – his expectation for recognition and for control of the system. Archer enjoyed four years at iAssemble in which he

⁵ The iAssemble organization and case example are completely fictional; any resemblance to a real organization or insider threat case is unintentional.

⁶ Personal predispositions are discussed in more detail later in this report.

had total control over the design and evolution of the company’s systems and networks. During that time, his expectation of continued control and prominence within the organization grew and became firmly entrenched. Archer’s personal predispositions exacerbated his sense of entitlement. Personal risk factors included Archer’s arrogant behavior in the workplace and his alcohol addiction problem. Archer was also under great personal stress due to family issues, which further amplified his disgruntlement at work.

Question#2: Why was Archer able to harm iAssemble after firing?

Discussion around this question typically focuses on how Archer accessed the system. Also relevant is the organization’s focus, prior to the attack, almost exclusively on the growth of the company with little or no recognition of the risks associated with that growth or with Archer’s actions in particular.

Another key question is why iAssemble fired Archer before cutting off all access. This naturally leads to the definition of an access path:

- *Access path*: a sequence of one or more access points that lead to a critical system.

An organization’s full awareness of access paths available to an insider is critical to being able to disable those access paths when needed. Figure 2 reflects the relationship between two variables: *Insider access paths unknown to org* and *Insider access paths known to org*. The flows between the two variables include

- *forgetting paths*: Management or the IT staff may forget about known paths, making them unknown. For example, a manager might authorize a software developer’s request for the system administrator password during a time of heavy development. If a formal list of employees with access to that password is not maintained, the manager could forget that decision over time. The manager may also simply resign from the organization, leaving no “organizational memory” of the decision to share the system administrator password.

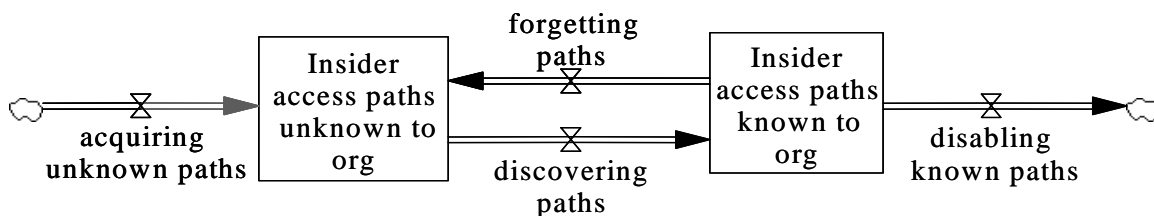


Figure 2: Access Paths Available to Insider

- *discovering paths*: Management or the IT staff can discover unknown paths, making them known. Access paths can be discovered by monitoring network traffic or by computer system account auditing. Monitoring network traffic allows discovering suspicious network traffic for further investigation. Account auditing allows discovering unauthorized accounts directly.

Insiders can acquire new paths unknown to the organization via the *acquiring unknown paths* flow. Finally, organizations can disable known paths via the *disabling known paths* flow. The

critical concept is that an organization may not know about all of the access paths each of their employees has to its critical systems. This leads naturally to the next question concerning prevention.

Question#3: What could iAssemble have done to prevent the attack?

One focus of this question is to understand the importance of actions or events that occurred, or conditions that existed, prior to the insider's attack. We define two key concepts as follows.

- *Behavioral precursor*: an individual action, event, or condition that involves personal or interpersonal behaviors and that precedes and is associated with malicious insider activity. Behavioral precursors from the iAssemble case include the insider's verbal threats and outbursts, visible stress at losing access and control, bottlenecks of projects, stress at home, and stress because of firing.
- *Technical precursor*: an individual action, event, or condition that involves computer or electronic media and that precedes and is associated with malicious insider activity. Technical precursors from the iAssemble case include the insider's developing a logic bomb, testing that logic bomb, and sharing passwords with a coworker.

Precursors may be both technical and behavioral in nature. For example, the sharing of passwords between Archer and a coworker facilitated the attack. The password sharing also opened an access path to the insider that the organization did not know about. In this particular case the organization may have closed this avenue of attack by

- prohibiting password sharing by policy, reinforced through periodic security awareness training
- instituting regular password changes, including administrator or other group accounts
- requiring all employees to change their passwords when Archer was fired.

Another focus of this question is to discuss how the organization could have used the knowledge of precursors to prevent the insider attack or otherwise mitigate the risk of attack. While some precursors can be prevented with minimal cost, others are better detected and responded to on a case-by-case basis. Behavioral precursors are often one of the first indicators of employee disgruntlement. If an organization is successful in identifying these precursors and taking measures to address them in a timely fashion, they might be successful in preventing attacks. This depends on perceptive management and targeted behavioral monitoring.

Technical precursors to an attack are even more serious and usually follow but may come in parallel with behavioral precursors. They may, by themselves, cause disruption in the organization's systems. They often indicate steps taken to set up a future attack on the organization's systems, possibly unbeknownst to the organization, such as creation of malicious code. Other technical precursors simply enable the insider to conceal his malicious acts. For instance, insiders often create fictitious (backdoor) accounts for their surreptitious entry to the system at a later date. This is an example of an access path that is not known by management. The organization needs to have technical monitoring in place to be able to detect such precursors at an early stage and they must take appropriate actions. While behavioral precursors, by themselves, are indicative of insider threat risk, the combination of technical and behavioral precursors indicates an even greater risk of insider attack.

Question#4: What should iAssemble do in the future?

This question requires that participants take a step back from the details of the particular scenario to describe what iAssemble should do in the future to ensure that the risk of insider IT sabotage is acceptably mitigated. Effective risk mitigation strategies should focus as much on understanding and reducing the impact of possible attacks as it does on preventing them in the first place. Organizational focus should be on those malicious acts with the largest potential impact to the organization.

Of course, organizations cannot prevent all malicious acts. They cannot even always monitor for all precursors equally. Difficult decisions must be made, especially regarding monitoring for technical precursors, due to their associated costs. Resources used to audit and monitor technical accounts and activities may divert effort from project deliverables. Comprehensive monitoring of all employees is often not cost effective. Organizations can, however, implement proactive monitoring and logging of all staff for a key set of precursors. When circumstances dictate an increased risk, they should engage in more detailed, targeted monitoring.

There are also difficult questions regarding which measures should be used to mitigate risks. Should the organization use technical measures like restricting access to curtail the risk of insider attack? Should it use non-technical measures like a warning or reprimand for concerning behaviors? An organization needs to take into account the effect of these technical measures on morale and productivity as well as risk. The organization also needs to be aware of access paths available to the insider, including indirect access paths like malicious code.

If choosing non-technical measures to reduce risk, the organization needs to consider positive intervention, such as constructive dialogue with employees or Employee Assistance Program (EAP) referrals, in addition to punitive techniques, such as reprimands or sanctions. For certain insiders, punitive techniques may increase the insider's disgruntlement to the point that they have little regard for future repercussions. Excessive or unreasonable monitoring within the workplace may make employees feel like they are being watched by "big brother." Low levels of trust within the workplace can de-motivate employees, creating an environment of low morale and low productivity. Of course different workplace cultures accept different levels of monitoring. Organizational management has to find the right balance for their particular organization between providing a trusting workplace environment and managing risks associated with insider IT sabotage.

3.2 General Observations

After discussion of the iAssemble case, workshop participants have a good understanding of some of the primary insider IT sabotage domain concepts. They have discussed them in terms of a specific, but fictional case. They often think about these concepts as they relate to their own personal and professional experience with employees within their own organizations. This generates interesting and often candid discussion of these concepts. It sometimes raises new issues that should be considered for future research.

This experience prepares the participants for a more general discussion on specific findings across all of the cases we have analyzed. We present these findings as a series of observations about common patterns seen.

- *Observation #1:* Most insiders had personal predispositions that contributed to their risk of committing IT sabotage.
- *Observation #2:* Most insiders who committed IT sabotage were disgruntled due to unmet expectations.
- *Observation #3:* In most cases stressors, including sanctions and precipitating events, contributed to the likelihood of insider IT sabotage.
- *Observation #4:* Behavioral precursors were often observable in insider IT sabotage cases but ignored by the organization.
- *Observation #5:* Insiders created or used access paths unknown to management to set up their attack and conceal their identity or actions. The majority of insiders attacked after termination.
- *Observation #6:* In many cases organizations failed to detect rule violations.
- *Observation #7:* Lack of physical and electronic access controls facilitated IT sabotage.

We describe each observation in terms of the percentage of cases that supports the observation, concrete observables from the cases that more concretely illustrate the observation, and at least one IT sabotage case example to further substantiate the observation. Presentation of the observations in this manner prepares the audience for the fundamental concepts on which the system dynamics model is based, to which we now turn our attention.

4 Model of the Insider IT Sabotage Problem

The *Insider Threat Study* investigated cases of actual insider attack. It therefore brought to light how the problem of malicious insider retribution arises and escalates within the organizational context. We learned much more about what organizations should not do than what they should do. Further research is needed into effective countermeasures, a research area that CERT is attempting to pursue.

This section describes the key elements of the insider IT sabotage problem that we saw in a majority of cases. The patterns embodied by the model were not seen in all cases but in a sufficient number of cases to raise concern. In the next section we will describe the measures that an organization can take to counter malicious insider actions based on our extended group's experience on the psychology of insiders as well as the managerial and technical aspects of organizational and information security.

For those readers familiar with the system dynamics, we emphasize that we do not use the traditional causal loop diagramming notation. As mentioned previously, workshop participants found the traditional notation using positive and negative signs to be confusing. Non-technical people generally were intimidated by the notation and technical people often read too much into the signs. In the following presentation, we use a more subtle notation of dashed arrows for negative influence and solid arrows for positive influence.

4.1 Insider Expectation Escalation

Employee disgruntlement was a recurring factor in the *Insider Threat Study* sabotage cases, predominately due to some unmet expectation by the insider. For example:

1. The insider expected certain technical freedoms in his⁷ use of the organization's computer and network systems, such as storing personal files, but was reprimanded by management for exercising those freedoms.
2. The insider expected to have control over the organization's computer and network system, but that control was revoked or never initially granted.
3. The insider expected a certain financial reward for his work, but bonuses were lower than expected due to the company financial status.

Figure 3 represents the escalation of expectation that often leads to insider disgruntlement. As shown in the lower left side of the figure, the insider's *personal predisposition* could lead to heightened expectation. This predisposition differs from one person to the next, and influences the rate that expectations rise and fall. Personal predispositions explain why some insiders carry out malicious acts, while coworkers that are exposed to the same conditions do not act maliciously. Personal predispositions can be recognized by certain types of observable characteristics (Band et al. 2006):

- Serious mental health disorders
- Social skills and decision-making
- A history of rule conflicts

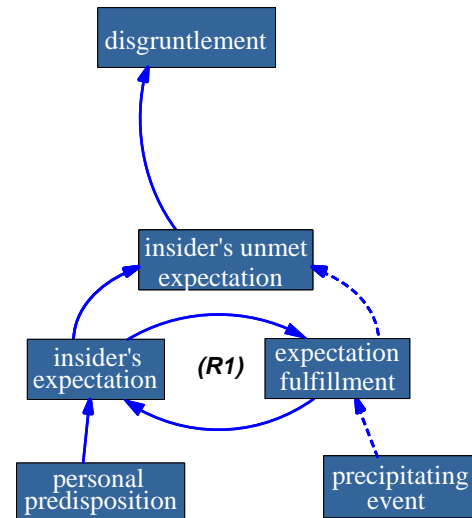


Figure 3: Expectation Escalation

The rise of expectations is influenced heavily by the *expectation fulfillment*. As illustrated in reinforcing loop (R1), with lax management controls the *insider's expectation* grows commensurate with the *expectation fulfillment*. As expectation grows and is fulfilled, expectation grows even more.

Lax management that permits increasing insider expectation can cause major problems later, especially if the insider is so predisposed. The trigger for those major problems, which we call the *precipitating event*, tends to be anything that removes or restricts the freedom or recognition to which the insider has become accustomed. In the iAssemble case, as in some cases in the *Insider Threat Study*, the trigger is the hiring of a new supervisor who enforces the organization's system usage policy. Other precipitating events include the insider being passed up for a promotion, sanctions by management, or termination of the insider.

4.2 Escalation of Disgruntlement

Often the first sign of disgruntlement is the onset of *behavioral precursors*, observable aspects of the insider's offline/social behavior inside or outside the workplace that might be deemed inappropriate or disruptive in some way. As shown in Figure 4a, the degree of disgruntlement influences the insider's exhibition of behavioral precursors, which can be discovered provided that the organization has sufficient *behavioral monitoring* in place. An organization's punitive response to inappropriate behaviors in the form of sanctions can be technical, such as restricting

⁷ Ninety-six percent of the insiders in the *Insider Threat Study* who committed IT sabotage were male. Therefore, male gender is used to describe the generic insider throughout this paper.

system privileges or right to access the organization's systems from home, or non-technical, such as demotion or formal reprimand.

The intended effect of sanctions, as shown in the balancing loop B1 of Figure 4b, is to prevent additional behavioral precursors. Feedback loop R2, however, shows that sanctions can have unintended consequences such as escalation of disgruntlement. Whether sanctions curb behavioral precursor activity or spur the insider to greater disgruntlement and disruption depends largely on the personal predispositions of the insider.

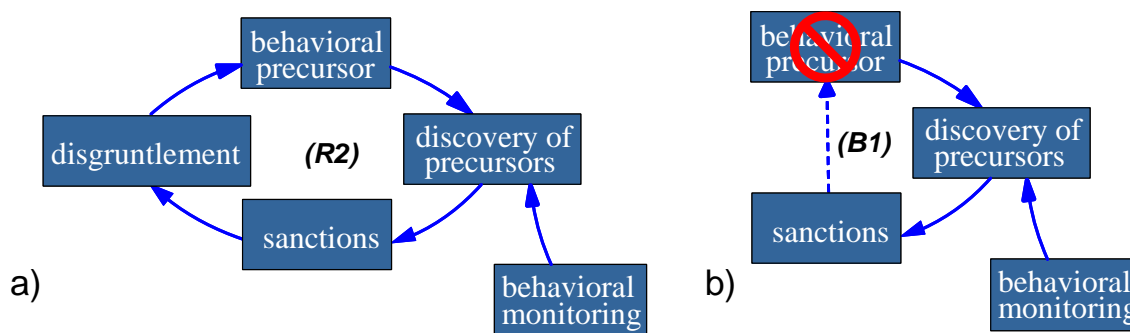


Figure 4: a) Typical Escalation of Disgruntlement b) Intended Effect of Sanctions

4.3 Attack Setup and Concealment

Given an insider with personal predispositions, unmet expectations can lead to increasing disgruntlement which, if left unchecked, can spur not just behavioral precursors but technical disruptions and attacks on the organization's computer and network systems. Prior to the actual attack, there are typically *technical precursors* - actions by the insider to either set up the attack (for example, installing malicious software programs) or to put in place mechanisms to facilitate a future attack (for example, creation of secret, unauthorized accounts to be used later for the attack). These technical precursors could serve as an indicator of a pending attack if detected by the organization.

Figure 5 depicts the influence that insider disgruntlement can have on the occurrence of technical precursors that could indicate a pending attack. Some of these actions also contribute to the damage potential of the attack. Examples include sabotage of backups and decreases in the redundancy of critical services or software. As shown in loop R3, insiders may also acquire access paths unknown to the organization. This increases the insider's ability to conceal their activity making it more difficult for the organization to discover the precursors. The feedback loop is reinforcing since the ability to hide their actions may embolden the risk-averse insider to continue, or even increase, their efforts to attack.

The extent to which insiders rely on unknown access paths to set up and execute their attack depends on their risk tolerance. Insiders who do not care whether they are caught, or insiders acting impulsively (often out of the passion of the moment), may use both known and unknown paths in their attack. Insiders who are particularly risk averse may only attack using access paths that are unknown to the organization. Of course, an insider may not know whether the organization is aware of a particular access path or not. Nevertheless, in either case, insiders generate technical precursors that suggest suspicious activity. Just as for behavioral precursors,

the detection of technical precursors depends on having sufficient level of technical monitoring in place.

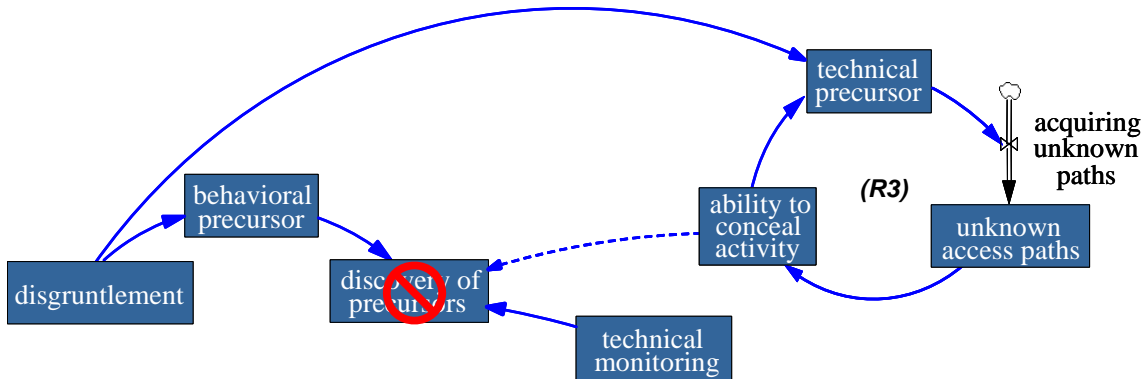


Figure 5: Technical Precursors due to Disgruntlement

4.4 The Trust Trap

In addition to insider predispositions and behaviors, organizational predispositions and behaviors can also influence an organization's exposure to malicious insider acts. Figure 6 depicts a trap in which organizations sometimes find themselves. We call this the Trust Trap and have described its role in previous models (Anderson et al. 2004, Cappelli et al. 2006, Band et al. 2006).

To understand the Trust Trap, we need to distinguish between the actual and perceived risk of insider attack. As shown in the top portion of Figure 6, actual risk depends on the behavioral and technical precursors exhibited by the insider. The risk of insider attack is only perceived by the organization to the extent that they discover those precursors, however.

A key factor in the Trust Trap is the organization's trust of the insider, as shown in loops R4a

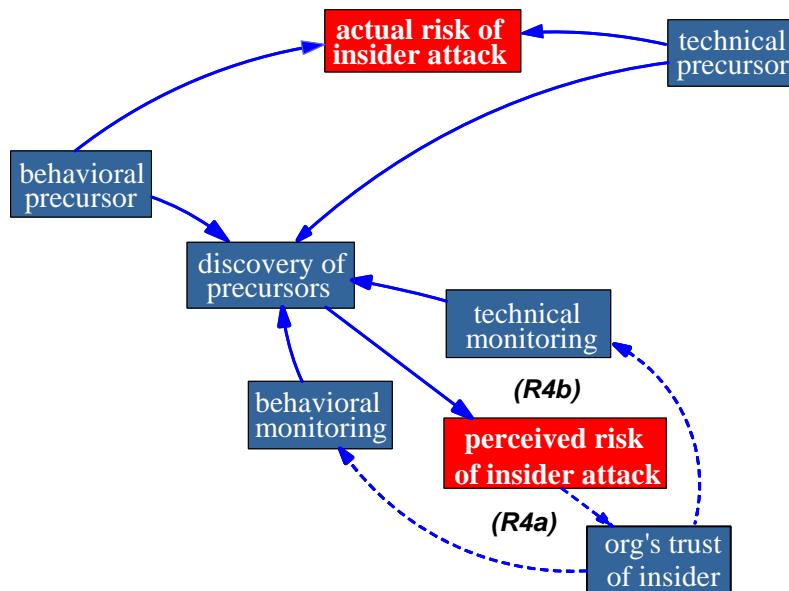


Figure 6: Trust Trap

and R4b. Clearly, there are good reasons why managers want to create a workplace in which individuals can trust each other and there is a good trust relationship between the organization and its employees, e.g., to increase morale and productivity. However, managers that strive to promote trusting workplace relationships sometimes shortcut essential behavioral and technical monitoring procedures. Lower levels of monitoring lead to undiscovered precursors, resulting in an overall lower perceived risk of attack. This false sense of security reinforces managers' trust in the individuals working for them. The cycle continues, with the organization's monitoring capability steadily deteriorating until a major compromise becomes obvious to all involved.

5 Possible Leverage Points for Addressing the Problem

The intent of our workshop is to communicate the severity of the insider threat problem and describe it using system dynamics models based upon empirical data. Although our research in CERT has focused on the insider threat problem, we would be remiss to leave participants with the impression that the organization is helpless to defend itself against someone from within. We can propose effective countermeasures based on our extended team's expert opinions in behavioral psychology and information security.⁸ Organizations, specifically management across the entire organization, should recognize and acknowledge the threat posed by insiders and take appropriate steps to mitigate malicious attacks. While it may not be realistic to expect that every attempt at insider IT sabotage will be stopped before damage is inflicted, it is realistic to expect that organizations can build resiliency into their infrastructure and business processes to allow the organizations to detect the attacks earlier, thereby minimizing the financial and operational impact.

This section of the report describes potential countermeasures that we believe could be effective in mitigating insider IT sabotage, based on expert opinions in our analysis of the problem.

5.1 Early Mitigation through Expectation Setting

First of all, managers should recognize the personal predispositions of their employees and understand the impact they can have on insider threat risk. Second, organizations should attempt to manage the expectations of employees to minimize unmet expectations. This can be achieved through communication between managers and employees (especially in the form of regular employee reviews), taking action to address employee dissatisfaction when possible, and consistent enforcement of policies for all employees so that individual employees do not come to feel that they are above the rules or that the rules are unjustly applied.

Figure 7 describes the influence *expectation setting* can have on the *insider's unmet expectations*. When the expectations of the insider are in line with the organization's practices and policies, unmet expectations are not an issue. However, if a precipitating event impacts expectation fulfillment, action by management to set expectations might decrease the level of unmet expectations. If the organization fails to reset expectations, the level of unmet expectations may continue to rise, causing disgruntlement on the part of the insider.

For example, the organization can attempt to lower the level of unmet expectations regarding system use and job responsibilities by a number of proactive countermeasures. For example:

⁸ The effectiveness of the countermeasures proposed in this section is not supported in the case data since we were rarely able to obtain that kind of data during the coding process.

- The organization institutes an acceptable use policy, describing the employee’s roles and responsibilities when using the organization’s computer and network system. The policy should be given to each employee as part of their orientation to the organization. As changes to the policy occur, employees need to be made aware of the changes and the impact to them. In addition, the policy should be consistently enforced for all employees so that no employees may feel that they are “above the rules.”
- Managers, in conjunction with Human Resources, can clearly define job responsibilities for each employee in the organization. Processes such as performance reviews can be used to check and set expectations periodically.

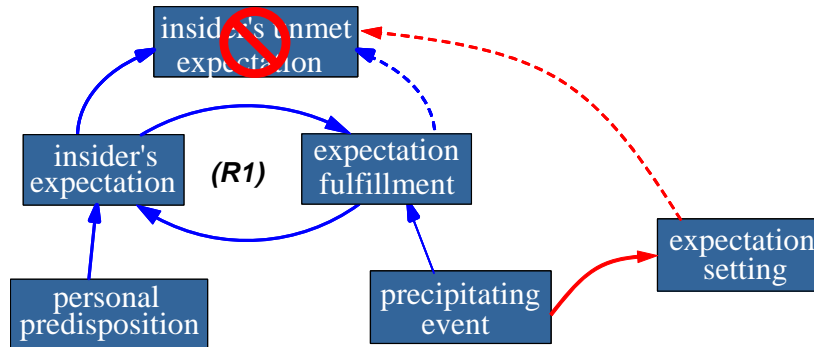


Figure 7: Early Mitigation through Expectation Setting

5.2 Handling Disgruntlement through Positive Intervention

As the organization discovers the behavioral precursors exhibited by the insider, they can employ positive intervention strategies to lower the disgruntlement of the insider. While the intent of employee sanctioning may be to reduce undesirable (non-)technical behaviors, it may backfire in some cases. Disgruntlement increases, leading to more disruptive behavior. Figure 8 describes the influence *positive intervention* strategies might have on the *disgruntlement* of the insider. When positive intervention is used, the disgruntlement might be reduced, eliminating additional behavioral precursors.

One positive intervention strategy is an Employee Assistance Program (EAP). EAPs are sometimes offered by organizations as an employee benefit, to assist employees in dealing with

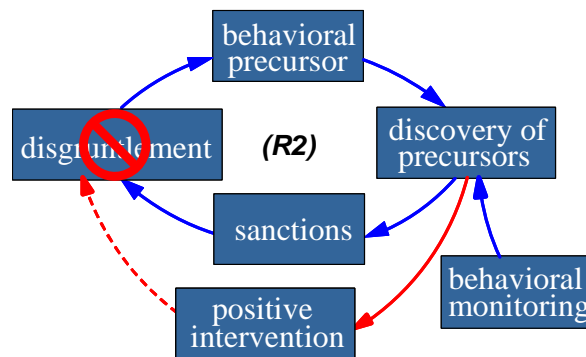


Figure 8: Handling Disgruntlement through Positive Intervention

personal or work-related issues that may affect job performance, health, and general well-being. EAPs can include counseling services for employees and/or their family members.

5.3 Targeted Monitoring

It is usually not practical for an organization to monitor every behavioral and technical action taken by each employee. However, a reasonable level of proactive logging of online activity across the organization's network provides data that can be monitored or audited for suspicious activity proactively, or targeted at a specific individual or individuals who have raised the suspicions of their managers.

Figure 9 describes the relationship between the *perceived risk of an insider attack* and the amount of *technical* and *behavioral monitoring* organizations institute.

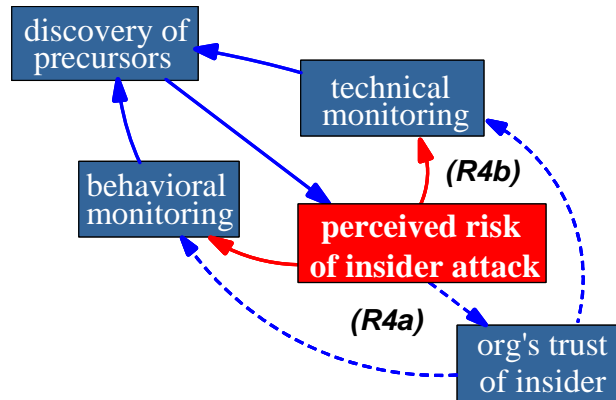


Figure 9: Targeted Monitoring

As the perceived risk of an insider attack increases, the amount of technical and behavioral monitoring should also increase. Enhanced monitoring should lead to discovery of precursor activity, enabling the organization to identify individuals at a higher risk for malicious behavior and implement more targeted individual monitoring. Band et al. (2006) identifies specific observable behaviors that should impact an organization's trust level.

5.4 Eliminating Unknown Access Paths

Access paths unknown to the organization provide a mechanism that can be used by the insider to facilitate a future attack, even following termination. In addition, unknown access paths can make it more difficult for the organization to attribute the attack to the insider. If the organization is unaware of the paths that can be used by an insider for attacks, the task of protecting itself is significantly more complex. Policies and procedures for tracking access paths, for example through account management and configuration management, can reduce the number of unknown access paths into the organization's critical infrastructure.

In the cases we examined, accounts that were secretly created by the insider or shared with other coworkers were access paths often used by the insider but unknown by management. Therefore, one practice for controlling unknown access paths is ongoing and thorough account management. Account management is a complex task, encompassing verification of new accounts, changes to account authorization levels, tracking who has access to shared accounts, and decommissioning of old accounts. Unfortunately, it takes a significant amount of time and resources for an organization to recover from obsolete account management practices.

Figure 10 emphasizes the importance of diligent tracking and management of access paths into the organization's system and networks. As tracking increases, the likelihood an organization will forget about the existence of specific access paths and who has access to them decreases. If precursor technical activity is detected, unknown access paths can be discovered and disabled, further reducing the number of unknown access paths available to the insider. Conversely, if technical precursors are not discovered then the insider can accumulate unknown access paths. In

many cases we examined, lack of tracking led to unknown access paths for the insider which were overlooked in the termination process, and later used by the insider to attack.

Finally, as the number of unknown access paths decreases, the ability to conceal malicious activity by the insider decreases. As the *ability to conceal* decreases, the discovery of technical precursors increases.

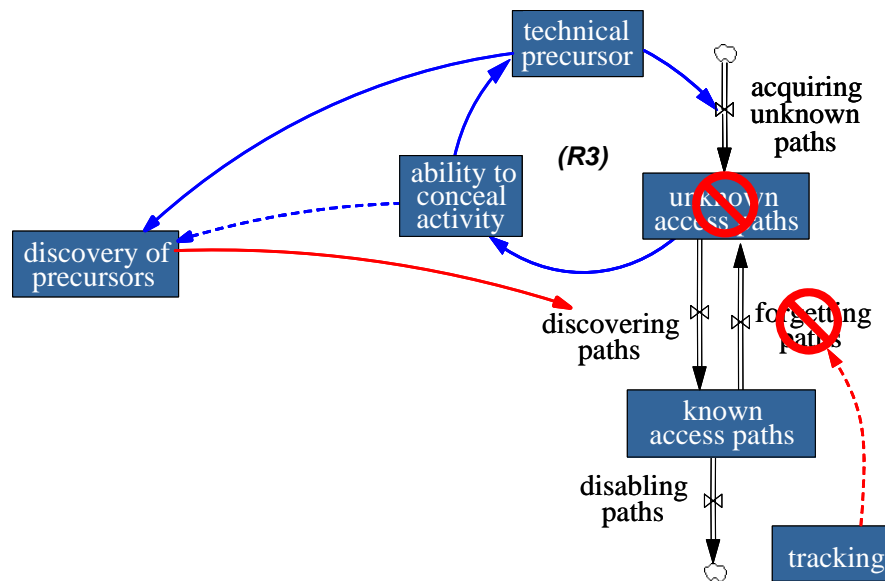


Figure10: Eliminating Unknown Access Paths

5.5 Measures upon Demotion or Termination

Termination or demotion was the final precipitating event in many cases we examined. It is important that organizations recognize that such precipitating events may cause the insider to take the necessary technical actions to set up and carry out the attack. A clearly defined process for demotions and terminations in combination with proactive IT best practices can reduce the insider’s ability to attack the organization.

Figure 11 reflects the steps that the organization can take to mitigate the insider IT sabotage risk following demotion and termination. Prior to the demotion or termination, the organization should be certain about what access paths are available to the insider. If the insider is to be terminated, the organization must disable all access paths prior to notifying the insider of the action. It is important to understand that if the organization has been lax in tracking and managing access paths, it could be too late to confidently demote or terminate an employee without fear of retribution.

When a demotion or termination occurs, the organization should analyze the roles and responsibilities of the new position as well as update the authorization levels and access controls, including role-based access. Some organizations in the cases we analyzed overlooked the change in privileges, allowing the employee to retain privileges from their previous position, giving them access to information beyond that needed for their new position.

Expectation setting during a demotion or termination can be a deterrent against future attacks. The employee should be clearly told what the acceptable use policy is regarding their new position, what their roles and responsibilities are in their new role, what their performance improvement plan is (if one exists), and that future monitoring and auditing will be implemented to measure job performance against individual and organizational goals and objectives.

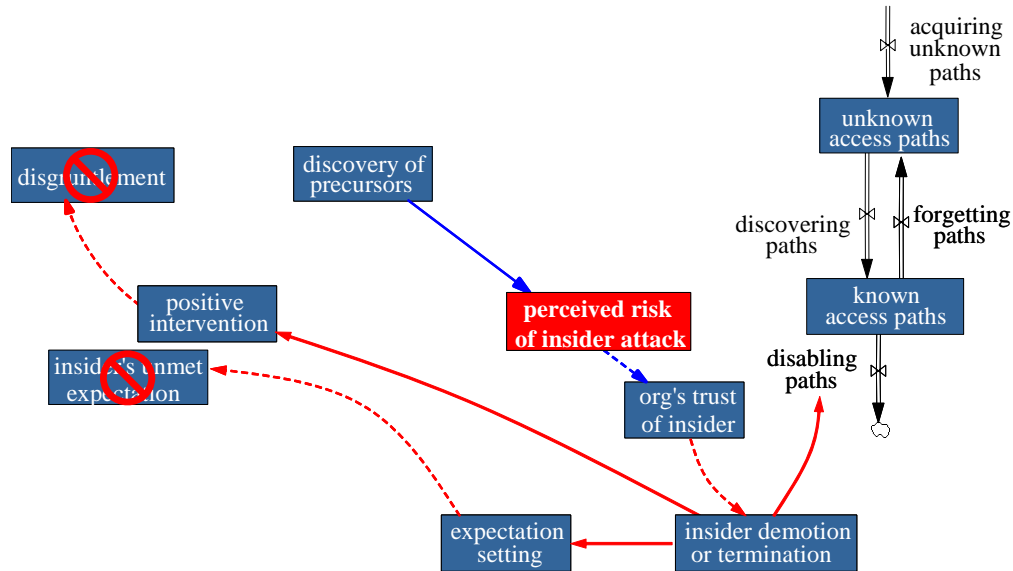


Figure11: Measures Upon Demotion or Termination

6 Conclusion

This paper focuses on how we have used system dynamics models to better communicate the nature of the insider threat problem within the context of an insider-threat workshop being developed at the CERT. This section describes the value of system dynamics modeling toward our better understanding of the insider threat problem in the first place, and the directions for our ongoing and future work.

6.1 Value of Modeling for Insight

We found that the system dynamics approach helped to structure and focus the team’s discussion. This was particularly important since members of the team, by necessity, came from the different disciplines of psychology and information security.

By identifying the primary variables of interest, the influences between these variables, and the feedback loops that are so important for understanding complex behavior, the team found itself able to communicate much more effectively. The group modeling process enabled the team to step back and consider the “big picture” at times and focus on individual concepts at other times. The rigorous notation helped identify commonalities to simplify the models and prevent misunderstandings that could have hindered progress otherwise. In addition, it was immensely valuable for each team member to be able to come away with the models that we developed after our group sessions and devote individual thought to each. It not only documented our progress

but helped us pick up from where we left off after a period of downtime and reflection on what we had accomplished. The models also provided a concrete target for validation through mapping to observables exhibited by the real-world cases.

Significant methodological and data challenges must be overcome before research on insider activity can be soundly prescriptive for mitigation policies, practices, and technology. However, we cannot overestimate the importance of looking at the total context of adverse insider behavior for understanding why these events happened and how they might be prevented in the future. By using the system dynamics approach we attempt to assess the weight and interrelatedness of personal, organizational, social, and technical factors as well as the effectiveness of deterrent measures in the workplace. Prospective studies of these phenomena will always be challenging because of low base rates. In the meantime, system dynamics modeling using available empirical data can bridge this methodological gap and translate the best available data into implications for policies, practices, and technologies to mitigate insider threat.

6.2 Ongoing and Future Research

Funded by the CyLab at the Carnegie Mellon University, our current project to model insider IT sabotage, called MERIT (for Management and Education of the Risks of Insider Threat), effectively combines psychological and technical findings from the joint CERT/U.S. Secret Service *Insider Threat Study* with other insider threat expertise from participating researchers. MERIT has greatly enhanced our ability to lead facilitated training on the risk of insider IT sabotage.

A follow-on project that started in September 2006, called MERIT-*Interactive*, builds upon the MERIT foundation to develop a stand-alone tool that can be used for more widespread training on insider threat risk mitigation. In collaboration with Carnegie Mellon's Entertainment Technology Center, we plan to use state of the art multi-media technologies to develop a compelling training simulation (which we call MERIT-*Interactive*, or MERIT_{IA} for short) that immerses players in a realistic business setting from which they (1) make decisions regarding how to prevent, detect, and respond to insider actions and (2) see the impacts of their decisions in terms of key performance metrics.

The MERIT-*Interactive* project will provide

- a stand-alone, multi-media training simulation for interactive and independent hands-on analysis of the effects of decisions regarding policies, practices, and technology on malicious insider activity based on the MERIT model for IT sabotage
- an effective means to communicate insider threat risks and tradeoffs, useful for both technical and non-technical personnel, from system administrators to corporate CEOs
- the state of the practice information regarding insider threats and effective countermeasures.

We are revising our previously developed simulation model to serve as a back-end engine for the tool. This approach should allow transferring our confidence in the insights provided by the model to confidence in the lessons being taught by MERIT_{IA}. Experiments will be needed to assess the extent which users of MERIT_{IA} are learning the important lessons within the insider threat domain. We believe that MERIT_{IA} will ultimately help decision-makers better understand insider threat risks and the effects of decisions on the promotion or mitigation of that risk. The

technology will empower organizations to develop comprehensive, efficient, and justifiable defenses to insider threats along with the organizational understanding and support needed to maintain a strong security posture over time.

To identify and better understand effective risk mitigators we also plan to collaborate with organizations to collect data on what policies, practices and technologies are being successfully implemented for insider threat prevention and detection. In addition, we will update our insider threat research by gathering and analyzing cases that occurred since the data collection phase of the *Insider Threat Study*.

7 Acknowledgements

CERT would like to thank the Army Research Office and Carnegie Mellon University's CyLab for funding this project.

CERT would also like to thank the following individuals for their collaboration on the MERIT model.

- Dr. Eric D. Shaw - Consulting & Clinical Psychology, Ltd., and a visiting scientist at CERT
- Dr. Stephen R. Band - Counterintelligence Field Activity – Behavioral Science Directorate
- Dr. Lynn F. Fischer – U.S. Department of Defense Personnel Security Research Center
- Dr. Elise A. Weaver – jointly as faculty at Worcester Polytechnic Institute and visiting scientist at CERT

Their expertise and experience in the psychological and social sciences areas have enabled a much richer treatment of the insider threat problem than would have otherwise been possible.

CERT also appreciates the work and dedication of the *Insider Threat Study* team members from CERT and the U.S. Secret Service, National Threat Assessment Center; without the study none of our follow-on insider threat research would have been possible.

Finally, Christopher Nguyen, a student at the Information Networking Institute of Carnegie Mellon University, and the anonymous reviewers for the ISDC 2007 provided many comments which improved both the content and presentation of this paper.

8 References

- Anderson, D.F.; Cappelli, D.M.; Gonzalez, J.J.; Mojtahedzadeh, M.; Moore, A.P.; Rich, E.; Sarriegui, J.M.; Shimeall, T.J.; Stanton, J.M.; Weaver, E.; and Zagonel, A. 2004. Preliminary System Dynamics Maps of the Insider Cyber-Threat Problem. *Proceedings of the 22nd International Conference of the System Dynamics Society*, July 2004. Available at <http://www.cert.org/archive/pdf/InsiderThreatSystemDynamics.pdf>.
- Band, S.R.; Cappelli, D. M.; Fischer, L.F.; Moore, A. P.; Shaw, E.D.; and Trzeciak, R.F 2006. "Comparing Insider IT Sabotage and Espionage: A Model-Based Analysis" Software Engineering Institute Technical Report CMU/SEI-2006-TR-026, Carnegie Mellon University, December 2006. <http://www.cert.org/archive/pdf/06tr026.pdf>.
- Cappelli, D. M.; Desai, A. G.; Moore, A. P.; Shimeall, T. J.; Weaver, E. A.; and Willke, B. J. 2006. "Management and Education of the Risk of Insider Threat (MERIT): Mitigating the Risk of Sabotage to Employers' Information, Systems, or Networks." *Proceedings of the 24th International System Dynamics Conference*. Nijmegen, Netherlands, July 2006. <http://www.albany.edu/cpr/sds/conf2006/proceed/proceed.pdf>.
- Keeney, M.M.; Kowalski, E.F.; Cappelli, D.M.; Moore, A.P.; Shimeall, T.J.; and Rogers, S.N. 2005. Insider Threat Study: Computer System Sabotage in Critical Infrastructure Sectors. *Joint SEI and U.S. Secret Service Report*, May 2005. Available at <http://www.cert.org/archive/pdf/insidercross051105.pdf>.
- Meadows, D. L.; Behrens, W. W.; Meadows D. H.; Naill, R. F.; Randers, J.; and Zahn, E. K. O. 1974. *Dynamics of Growth in a Finite World*. Cambridge, MA: Wright-Allen Press, Inc..
- Melara, C.; Sarriegui, J.M.; Gonzalez, J.J.; Sawicka, A.; and Cooke, D.L. 2003. A system dynamics model of an insider attack on an information system. *Proceedings of the 21st International Conference of the System Dynamics Society* July 20-24, New York, NY, USA.
- Moore, A.P.; Joseph, H.G.; Trzeciak, R.F.; Cappelli, D.M. 2007. Instructional Case of Insider IT Sabotage: An Instructor's Manual, in preparation.
- Naumes, W.; and Naumes, M.J. 1999. *The Art & Craft of Case Writing*. Thousand Oaks, California: SAGE Publications.
- Rich, E.; Martinez-Moyano, I.J.; Conrad, S.; Cappelli, D.M.; Moore, A.P.; Shimeall, T.J.; Andersen, D.F.; Gonzalez, J.J.; Ellison, R.J.; Lipson, H.F.; Mundie, D.A.; Sarriegui, J.M.; Sawicka, A.; Stewart, T.R.; Torres, J.M.; Weaver, E.A.; and Wiik, J. 2005. Simulating Insider Cyber-Threat Risks: A Model-Based Case and a Case-Based Model. *Proceedings of the 23rd International Conference of the System Dynamics Society*, July 2005.
- Sterman, J.D. 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. New York, NY: McGraw-Hill.

Appendix A: System Dynamics Background

System dynamics is a method for modeling and analyzing the holistic behavior of complex problems as they evolve over time. System dynamics has been used to gain insight into some of the most challenging strategy questions facing businesses and government for several decades. System dynamics provides particularly useful insight into difficult management situations in which the best efforts to solve a problem actually make it worse. Examples of these apparently paradoxical effects include the following (Sterman 2000).

- Low-nicotine cigarettes, supposedly introduced to the benefit of smokers' health, that only result in people smoking more cigarettes and taking longer, deeper drags to meet their nicotine needs
- Levees and dams constructed to control floods that only produce more severe flooding by preventing the natural dissipation of excess water in flood plains

The *Insider Threat Study* found that intuitive solutions to problems with employees often reduce the problem in the short term but make it much worse in the long term. For example, employee termination might solve an immediate problem, but it may also lead to long-term problems for the organization if the insider has the technical means to attack the system following termination. System dynamics is a valuable analysis tool for gaining insight into long-term solutions and for demonstrating their benefits.

A powerful tenet of system dynamics is that the dynamic complexity of problematic behavior is captured by the underlying feedback structure of that behavior. We decompose the causal structure of the problematic behavior into its feedback loops to understand which loop is strongest (i.e., which loop's influence on behavior dominates all others) at particular points through time. We can then thoroughly understand and communicate the nature of the problematic behavior and the benefits of alternative mitigations.

System dynamics model boundaries are drawn so that all the enterprise elements necessary to generate and understand problematic behavior are contained within them. This approach encourages the inclusion of soft (as well as hard) factors in the model, such as policy-related, procedural, administrative, or cultural factors. The exclusion of soft factors in other modeling techniques essentially treats their influence as negligible, which is often not the case. This endogenous viewpoint helps show the benefits of mitigations to the problematic behavior that are often overlooked, partly due to a narrow focus in resolving problems.

In this project we rely on system dynamics as a tool to help understand and communicate contributing factors to insider IT sabotage and espionage threats and implications for various mitigation strategies and tactics. It is tempting to use the simulation of the model to help predict the effect of mitigation strategies, but what is the nature of the types of predictions that system dynamics facilitates? Dennis Meadows offers a concise answer by categorizing outputs from models as follows (Meadows et al. 1974).

1. Absolute and precise predictions (Exactly when and where will the next cyber attack take place?)
2. Conditional precise predictions (How much will it cost my organization if a cyber-attack occurs?)

3. Conditional imprecise projections of dynamic behavior modes (If a bank mandates background checks for all new employees, will its damages from insider fraud be less than they would have been otherwise?)
4. Current trends that may influence future behavior (What effect will current trends in espionage have on national security in five years?)
5. Philosophical explorations of the consequences of a set of assumptions, without regard for the real-world accuracy or usefulness of those assumptions (If a foreign country succeeds in human cloning, how would this affect the United State's risk of espionage?)

Our models and system dynamics models, in general, provide information of the third sort. Meadows explains further that “this level of knowledge is less satisfactory than a perfect, precise prediction would be, but it is still a significant advance over the level of understanding permitted by current mental models.”

As described in the main body of this paper, we have modified the system dynamics causal loop diagram notation to be more suitable for the expected participants of our workshop. Arrows still represent the pair-wise influence of the variable at the source of the arrow on the variable at the target of the arrow, but their look indicates how they should be interpreted:

- Roughly, a *solid* arrow indicates a *positive* influence - that the value of the source and target variables moves in the *same* direction.⁹
- Roughly, a *dashed* arrow indicates a *negative* influence - that the value of the source and target variables moves in the *opposite* direction.¹⁰

As mentioned, dynamically complex problems can often be best understood in terms of the feedback loops underlying those problems. There are two types of feedback loops: *balancing* and *reinforcing*.

- Balancing loops (labeled *B#* in the figures) describe the system aspects that oppose change, tending to drive organizational variables to some goal state. In other words, balancing loops tend to move the system to an equilibrium state even in the face of change. The behavior of a thermostat is an example of a balancing loop. It continually changes the air flow into a room based on the temperature of the room, with the goal of maintaining an equilibrium temperature.
- Reinforcing loops (labeled *R#* in the figures) describe the system aspects that tend to drive variable values consistently upward or consistently downward. In other words, reinforcing loops can “spiral out of control.” A flu epidemic is an example of a reinforcing loop. It spirals out of control as more and more people contract the flu.

⁹ More formally, a *solid* arrow indicates that if the value of the source variable increases, then the value of the target variable increases above what it would otherwise have been, all other things being equal. And, if the value of the source variable decreases, then the value of the target variable decreases below what it would otherwise have been, all other things being equal.

¹⁰ More formally, a *dashed* arrow indicates that if the value of the source variable increases, then the value of the target variable decreases below what it would otherwise have been, all other things being equal. And, if the value of the source variable decreases, then the value of the target variable increases above what it would otherwise have been, all other things being equal.

The type of a feedback loop is determined by counting the number of negative influences along the path of the loop. An odd number of negative influences indicates a balancing loop, and an even (or zero) number of negative influences indicates a reinforcing loop.

System dynamics models are described as a sequence of feedback loops that characterize how the problem unfolds over time. Each feedback loop describes a single aspect of the problem. Multiple feedback loops interact to capture the complexities of the problem domain.

Appendix B: The Insider IT Sabotage Training Case¹¹

Introduction

Chris Eagles, president of the computer systems sales company *iAssemble*, felt like he had just been hit by a Mack truck.

A partner in the company, Caroline Thompson, explained to him that something had just wiped out their system configuration and assembly programs. “And to top it off,” she continued, “the only backups were given to Ian Archer before he was fired and we haven’t seen them since. Given the circumstances of Ian’s departure, we suspect that he might be responsible.”

“I just can’t believe that Ian would do something like that,” said Chris. “He’s been with the company since the beginning; he wrote most of those programs himself, for crying out loud!”

Chris paused and looked back at Caroline, “What the heck do we do now?”

Background

iAssemble sold computer systems directly to customers, building each system made-to-order and offering competitive prices. *iAssemble* had been doing extremely well and conducted an initial public offering (IPO) in 2001, after which its stock doubled.

Eagles started the company in 1997 with his friend Caroline Thompson, who is now the Chief Technology Officer (CTO). The company has had success hiring experienced managers and employees since the beginning.

Ian Archer was among the few employees who had been with *iAssemble* since its establishment. Archer started out as computer specialist and technical assistant to the two original founders, Eagles and Thompson. When hired, Archer held certifications in personal computer (PC) hardware maintenance and operating system administration. Although he did not possess a college degree, with hard work and dedication, he became the sole system administrator at *iAssemble*, a position he held for four years. He also built the software that ran the computer system assembly machinery pretty much from the ground up. This software was the foundation for automating the PC assembly processes that allowed *iAssemble*’s rapid growth.

iAssemble grew at an increasing rate. Recognizing the need for qualified personnel, Eagles and Thompson began to hire experienced system administrators who could also function as project managers. Lance Adams was hired as lead system administrator because of his education and experience, and, James Allen was hired as a junior system administrator to share Archer’s growing systems administration workload and responsibilities.

Archer was assigned to mentor Allen and ensure his smooth assimilation within the company. Archer and Allen worked on several projects together. Archer respected Allen’s abilities, finding him to be nearly as technically competent as himself. The two of them got along fine, but Allen began to notice that Archer was becoming increasingly short-tempered over a period of several months.

¹¹ The *iAssemble* organization and case example are completely fictional, any unintentional resemblance to a real organization or insider threat case is unintentional.

One day after Archer's moodiness started annoying him, Allen decided to find out what the problem was. "Hey, what's bugging you Ian?" probed Allen.

As if he had been waiting for someone to ask that question, he dumped on Allen. "James, I can't take it any longer. That idiot Adams thinks he knows how to run these networks better than me. I know these systems inside and out. The changes that he is suggesting will bring the network to its knees. They've got me on these piddly projects while they are destroying the foundations that I laid for iAssemble."

"You should be managing these networks, Ian," suggested Allen. "Why didn't they make you lead admin?"

"I have no idea, but who wants to be pushing papers all day long, anyway" Archer interrupted. "Adams is perfectly suited to that, but he doesn't know the first thing about running these networks. They simply don't appreciate what I do around here and some day they may just regret it."

Archer's disgruntlement grew and it became obvious in his hostile dealings with coworkers. He even bottlenecked projects on purpose on several occasions, stalling his work to ensure Adams and the team missed project milestones. Archer received a written warning from Thompson after several co-workers formally complained. Enraged by this, he had a heated argument with a team member who quit the very next day, citing Archer as the reason for his resignation. Thompson sent Archer a letter of final warning that put him on probation for his conduct – any more such problems would result in his immediate termination from iAssemble.

This seemed to resolve the situation, at least for a while. During the subsequent months, iAssemble continued to thrive. With a whopping 68% growth in sales over the previous quarter, iAssemble was forced to hire additional people. The staff adopted a "do whatever it takes" attitude to their job in order to keep up with the demands placed on them due to the growth. One staffer described it as follows:

"We were one lean coding machine in those days. We had to be to extend the systems to support the company's amazing growth. Ian thought of and implemented the idea to centralize the core software on a central server to coordinate updates more efficiently. We made vast improvements to the flexibility and sophistication of the assembly programs over a very short period of time. And the extensions worked well with very few glitches. Of course, we had to cut some corners, giving people access when and where they needed it to make things happen. If something did not contribute to extending the systems, it just did not seem worth doing. This was how we were able to accomplish as much as we did."

Unbeknownst to management, during these months Archer was busy developing and testing a logic bomb that would delete all of the files on the central server. He did the testing and some of the development on his office desktop machine to make sure that it would really work. He also planted a backdoor account, with administrator privileges, on the main machinery server that provided him with unconstrained access from home just in case he needed it.

Archer tried to get along with his coworkers, knowing that he would eventually get even. But he viewed most of his coworkers as incompetents; he just could not help "letting loose" on them every once in a while. He had already started looking for another job, one where his abilities were recognized and valued, so he felt that he would not be around iAssemble much longer.

The Final Weeks

Management decided that they needed to deal with Archer's lingering performance problems. In a meeting with Eagles and Thompson, Adams complained, "Ian is an arrogant jerk. He harasses and bullies his coworkers, treating them like they are dirt under his feet. Larry, the new programmer that we hired a few months ago, suspects that Ian is messing with the code that he is developing to make him look bad. Most of the staff walks on eggshells around him. We've got to do something."

"How big a hole is it going to leave in development and operations if we fire him?" asked Eagles.

"Virtually none," replied Thompson, "with all of our hiring and aggressive training programs over the past few months, the rest of the staff is well up to speed on how things run and the directions that we are going. Both James and I think that we'd be a whole lot better off without him."

"OK," replied Eagles, "Caroline, you take care of this yourself. Make sure to coordinate with James to be darned sure you cut off his access before you let Ian know. I feel bad about this – Ian has been with us since the beginning, but he has brought this on himself. So let's make it happen. How soon should we do it?"

"The sooner the better in my book." said Thompson, "I'll schedule to meet with him this morning. Lance, you disable his access while I'm meeting with him, and I'll have security escort him from the building after the meeting."

Eagles' comment about disabling Archer's access had left Adams with concerns. Security practices at iAssemble had been less than rigorous lately, with the push to get the new software out. But Adams decided not to voice them at the meeting and later that day on July 10, 2001, Archer was fired, his access was disabled, and he was escorted from the building just as they had planned. Passwords for all shared accounts, including the system administrator accounts, were changed while Thompson was meeting with Archer.

Unfortunately, iAssemble managers were not aware that Allen had shared his password with Archer months earlier in order to make the development process easier. Archer went home the night he was fired and successfully logged into Allen's account. He then used his backdoor account on the machinery server to plant the logic bomb. He set it to go off one week later.

After the logic bomb detonated, Thompson was in Eagles' office explaining that their critical software had been wiped out. Eagles was puzzled as to how that was possible in light of the monitoring, policies, and security practices in place at iAssemble. After numerous hours of investigation, the system logs were used to trace the access of the machinery server to Allen's account. The evidence pointed to James Allen as the saboteur. Allen claimed that he was not responsible for the deleted software and explained that he had given the password to his machine to Archer when they worked together. According to Allen, it had been such a long time ago that it had slipped his mind.

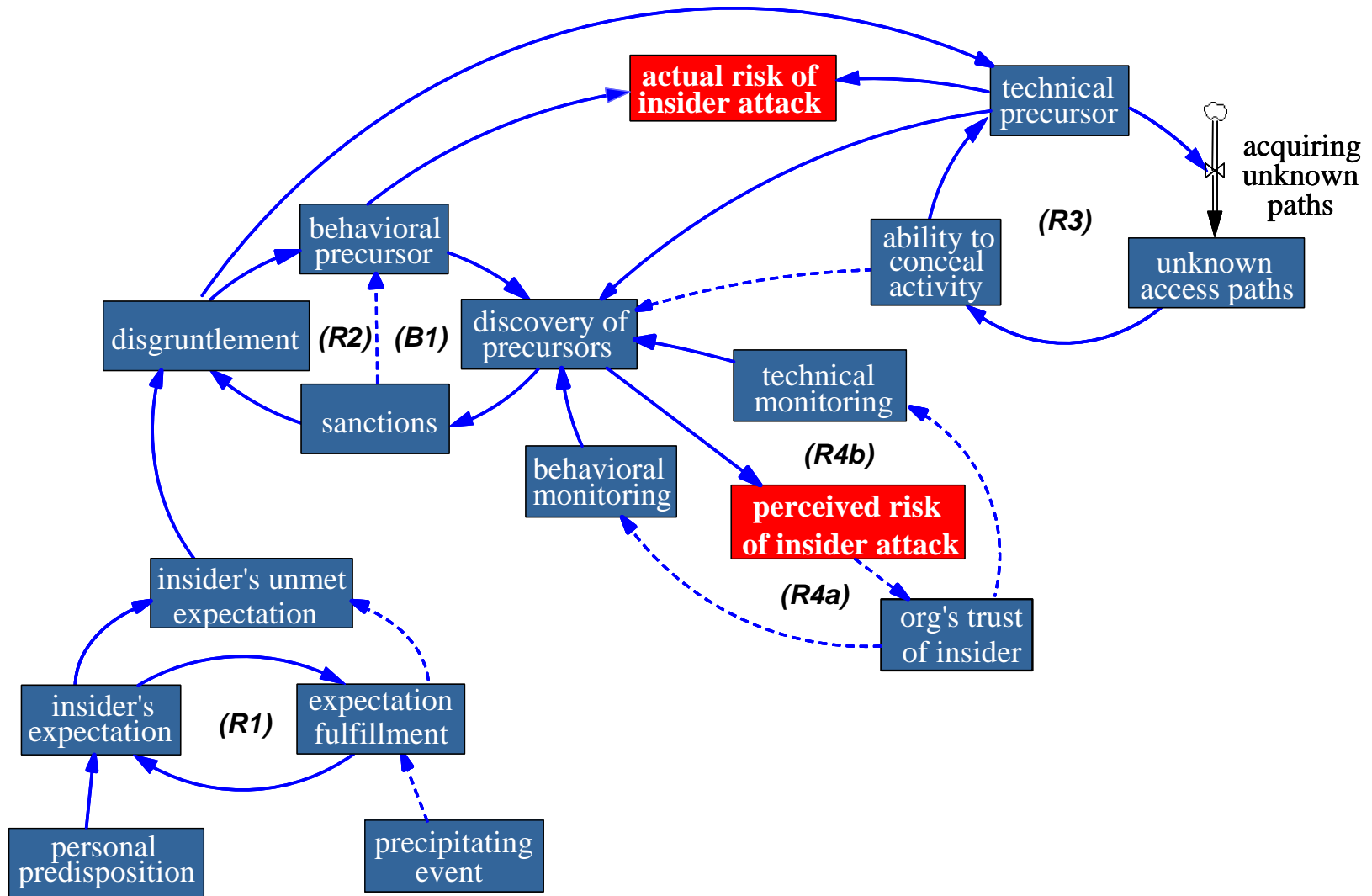
Management decided to call in law enforcement. Forensics analysis revealed that Allen's account was accessed remotely from Archer's home. Analysis of Archer's computer showed that he tested the logic bomb four times over a period of three months. When questioned, Ian Archer

continued to maintain his innocence, even though the evidence against him was substantial. Investigation into Archer's background revealed that his father had been suffering from lung cancer over the last year and that he had recently lost his driver's license due to a conviction for driving under the influence of alcohol.

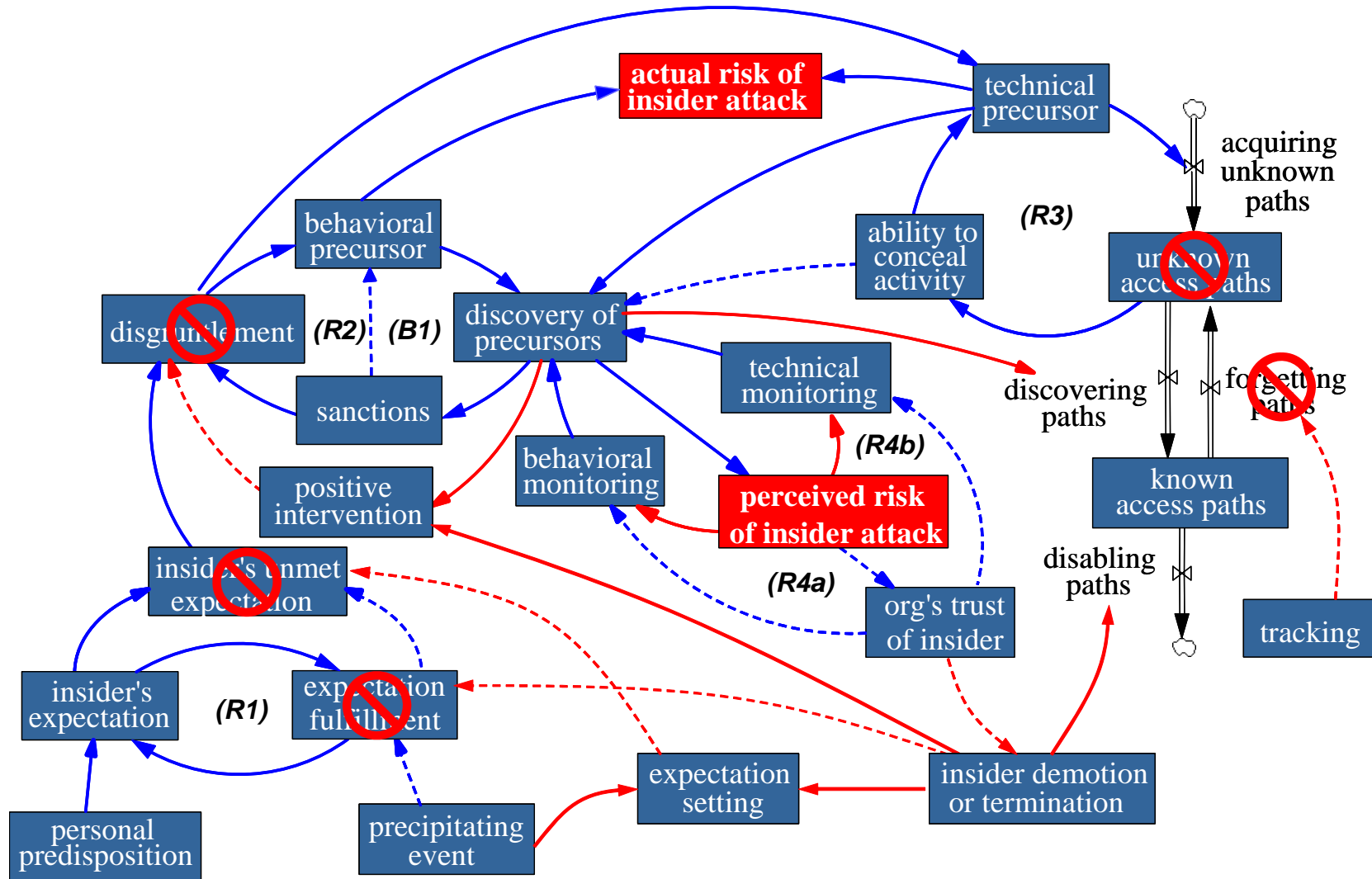
In a meeting with Thompson and Adams, Eagles exploded, "We know who did it, but how do we recover from something like this? It will take months to recover operations even close to what we had. When this gets out stockholders are going to demand a detailed explanation."

Slamming his fist on his desk, Eagles demanded, "We must not only understand how this happened, but why, and make sure it does not happen again!"

Appendix C: Model of the Insider IT Sabotage Problem



Appendix D: Insider Sabotage Mitigating Measures



Appendix E: Glossary of Terms

Term	Definition
Action	An <i>individual action</i> or an <i>organization action</i>
Access Path	A sequence of one or more access points that lead to a critical system
Behavioral precursor	An <i>individual action, event, or condition</i> that involves personal or interpersonal behaviors and that precedes and is associated with <i>malicious insider activity</i> ; inferable from a set of <i>observables</i>
Condition	A state of existence inside or outside the organization
Event	A happening that occurs outside the organization.
Individual action	An act of a <i>malicious insider</i>
Insider	A current or former employee or contractor of an organization
Insider IT sabotage	An <i>insider's</i> use of IT to direct specific harm at an organization or an individual
Malicious insider	An <i>insider</i> who engages in <i>malicious insider activity</i>
Malicious insider activity	Activity associate with <i>insider IT sabotage</i> (for the purposes of this paper)
Observable	An <i>action, event, or condition</i> that could be observed
Organization action	An act of an organization, as a whole, or by a non-malicious insider within the organization.
Personal predisposition	A characteristic historically linked to a propensity to commit <i>insider IT sabotage</i> ; inferable from a set of <i>observables</i>
Saboteur	<i>Insider</i> who commits or attempts to commit <i>insider IT sabotage</i>
Technical precursor	An <i>action, event, or condition</i> that involves computer or electronic media and that precedes and is associated with <i>insider IT sabotage</i> ; inferable from a set of <i>observables</i>
Unmet expectation	An unsatisfied assumption by an individual that an <i>organization action</i> or <i>event</i> will happen, or a <i>condition</i> will exist; inferable from a set of <i>observables</i>