A Forensic Framework for Incident Analysis Applied to the Insider Threat

Clive Blackwell

Department of Computer Science,
Oxford Brookes University,
Oxford OX33 1HX. UK
C.Blackwell@brookes.ac.uk

Abstract. We require a holistic forensic framework to analyze incidents within their complete context. Our framework organizes incidents into their main stages of access, use and outcome to aid incident analysis, influenced by Howard and Longstaff's security incident classification. We also use eight incident questions, extending the six from Zachman's framework, to pose questions about the entire incident and each individual stage. The incident analysis using stage decomposition is combined with our three-layer incident architecture, comprising the social, logical and physical levels, to analyze incidents in their entirety, including human and physical factors, rather than from a technical viewpoint alone. We demonstrate the conjunction of our multilayered architectural structure and incident classification system with an insider threat case study, demonstrating clearly the questions that must be answered to organize a successful investigation. The process of investigating extant incidents also applies to proactive analysis to avoid damaging incidents.

Keywords: Forensic incident framework, incident questions, insider threat, Zachman's framework.

1 Introduction

1.1 Rationale and Previous Work

We first explain the rationale for our forensic incident framework before moving on to describe its design in more detail. Security incident classifications often give subjective and incomplete representations of systems and incidents by focusing on logical aspects rather than the entire context. We need to model all possible evidence in incident investigation to meet the goals of cause attribution, enabling recovery, fixing weaknesses and disciplining the perpetrator.

We consider incidents within a wider context and from multiple perspectives to aid a broader and deeper investigation. The focus is extended from misused computer systems to their wider social, legal, personal, organizational, physical and environmental contexts. Most incident classification schemes do not fully analyze the progression of incidents through their various stages or the relationships between the involved entities. Our framework allows the decomposition of complex incidents into their atomic stages along with their causes and effects. This is crucial, because

P. Gladyshev and M.K. Rogers (Eds.): ICDF2C 2011, LNICST 88, pp. 268–281, 2012.

evidence about incident events and their timeline may be partial and indirect, and we may have to infer missing events from the hypothesized incident pattern.

We developed a new incident architecture that considers damaging incidents in their entirety, rather than as logical incidents alone [1]. Our three-layer architecture comprises the social, logical and physical levels, inspired by the OSI seven-layer network model [2]. This allow a holistic and comprehensive forensic analysis taking account of the entire context, including human and physical factors, rather than from a technical viewpoint alone, to observe, analyze and prove incident causality.

1.2 The Zachman Framework

The Zachman framework [3] is a complex model for designing enterprise computing architecture that tries to capture and organize information about every aspect of an organization relating to its computing requirements. Zachman provides a two-dimensional grid, where six questions are posed to describe the different aspects of the system, which are answered for each of five conceptual levels, leading to a five by six grid. These six questions are who, what, why, when, where and how. We use these as incident questions within our forensic framework to guide the investigative process. Interestingly, the Department of Justice asks five of the six questions (omitting *why* for some reason), within the analysis phase of their Digital Forensics Analysis Methodology [4]. We extend Zachman's framework with two more questions: *with what* is the means of attack, and *to what* is the target. The eight questions are answered for the entire incident and each stage to help establish comprehensive incident analysis.

1.3 Howard and Longstaff's Security Incident Classification

Our framework organizes incidents into stages with different purposes, actors, scope and effects, influenced by Howard and Longstaff's security incident classification [5, 6]. We give a very brief summary of our incident ontology [1]. We started with Howard and Longstaff's taxonomy for network security incidents in figure 1 that shows the different entities involved in incidents and their relationships. The categories are attacker, tool, vulnerability, action, target, unauthorized result and objectives. The attacker uses a *tool* to perform an *action* that exploits a *vulnerability* on a *target* causing an *unauthorized result* that meets its *objectives*.

2 Forensic Incident Framework

We extended Howard and Longstaff's classification [6] to include the social and physical incident aspects, which allows more detailed and comprehensive incident analysis. We used our three-layer incident architecture comprising the social, logical and physical levels to consider incidents in their entirety, rather than as logical incidents alone [1].

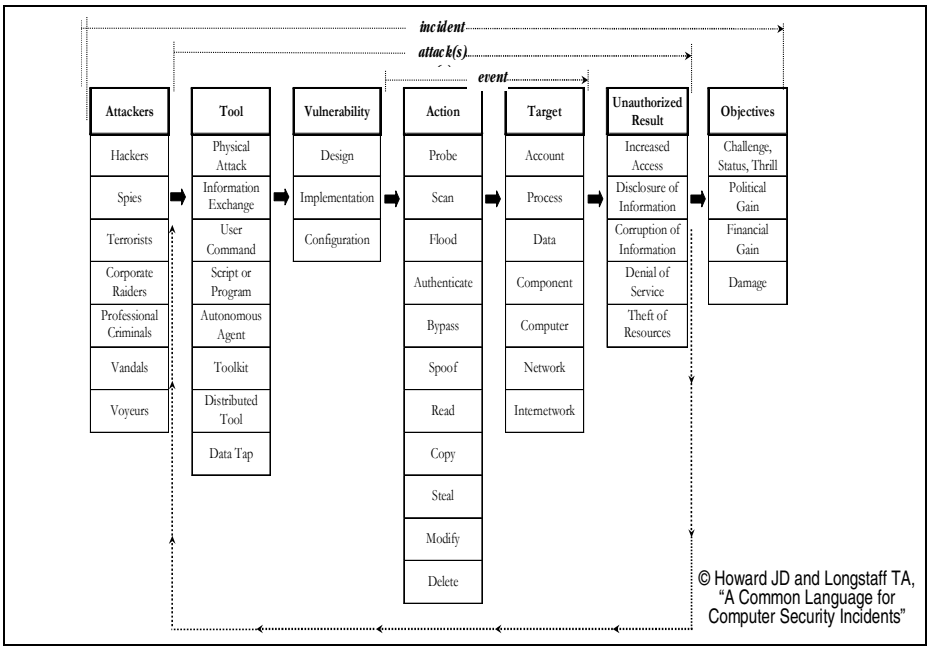

Fig. 1. Howard and Longstaff's Security Incident Taxonomy

The social level is the location for incident perpetrators and their intangible attributes such as motivation, and allows differentiation between real world actions and the resulting effects on people and organizations. The physical level is also significant for computer incidents, as many combine logical and physical aspects, and any computational activity is ultimately performed physically. This allows a holistic forensic analysis of the entire incident.

We divided the incident into its individual stages, as each stage has a particular purpose in supporting the incident goals, possibly performed by different stage actors with differing motivations and abilities. The three main active stages in an incident are system access, target use and incident outcome, possibly with optional stages such as initial reconnaissance and subsequent use of acquired resources. Altogether, there are five stages including the incident prologue and epilogue along with the three active stages.

Howard's incident model, as many others, does not mention system recovery, third party victims, or the perpetrator leaving the system and its subsequent use of acquired resources, but analysis of the final outcome stage is crucial to the investigatory process. Classifying the incident into stages aids analysis compared to considering isolated events.

In an active *stage* of an incident, the *actor* (*perpetrator* or its *agent*) performs an *action* with an *immediate effect* on a *target* (central rows in table 1). This ultimately causes the social-level *ultimate effect* that meets the perpetrator's psychological, functional or financial *objective* at the expense of the *ultimate target*, which is an organization or person. We distinguish between the *immediate effect* at lower levels on the targeted system and resources, and the *ultimate effect* on the *ultimate target*.

The incident is only successful if the perpetrator's objective is met by the ultimate effect, because resource access is usually only a means to the goal. For example, the ultimate objective for an insider causing sabotage is usually psychological satisfaction from revenge, which cannot be captured by only considering the low-level damaging physical or logical effects.

We link the incident classification with the eight incident questions to help organize the investigation. A complete table has headings for the incident entities, processes, purposes and outcomes, along with the incident questions, with six of the questions as subheadings of both an incident and stage column. All six questions are shown for each stage, but not for the entire incident for space reasons. The overall incident table would also include the incident prologue and epilogue leading to a five-stage model including the three active stages of access, use and outcome.

When and *where* is conceptually within the table giving a logical incident spacetime. The *when* question is answered implicitly from the temporal progression of the incident from left to right, except within the middle columns representing individual stages, when the outcome from one stage may move backwards to be used as an input to a later stage.

The different stages of access, use and outcome can be shown sequentially, rather than overlaid upon each other, for greater convenience when investigating complex incidents. We raise information collected about the incident to evidence at the social level by answering the incident question in every column.

3 Insider Attack Classification

3.1 Types of Insider Attack

We classify attacks into their ultimate effects on the organization of sabotage, fraud and theft to satisfy the goals of the attacker, which is based on the classification used in the second CERT guide to insider threats [7]. This is slightly different from the classification used in the current third guide, where the three classes are sabotage, financial gain and business advantage, which focus on the purpose (why) of the attack for the perpetrator rather than the actions involved (how) [8]. Our classification system considers all incidents, so our classes include physical as well as logical ones.

Our classification is comprehensive as it models incidents at all levels for holistic analysis, whereas the CERT and SEI research only discusses incidents involving logical computer resources, because the focus of their research is to understand, evaluate and manage the logical insider threat. Many incidents involve social and physical aspects along with the logical aspects, so our structured multilevel incident analysis helps to uncover the organizational and physical weaknesses and effects of their breach. For example, the perpetrator in the iAssemble case study discussed later installed a logic bomb (logical action) to delete the software controlling the production line so that the computers could not be manufactured (physical effect) with a resulting loss of income (organisational social-level effect).

The undesirable effects on the organization at the social layer, such as losing money, reduced business efficiency, failing to provide services, reputational damage and loss of customers should be highlighted rather than focusing on protecting assets. Then, there is the possibility of meeting the organizational goals even when the

protection is defeated, by taking recovery measures within a wider organizational scope than the immediately affected resources. This requires clear understanding of the relationship between the underlying services provided by resources and the business goals, which our model helps to elucidate.

Sabotage and damage usually cause the loss of availability and integrity of the targeted resources with possible consequential effects on the ability of the organization to perform its normal business activities. We should distinguish the intent of the perpetrator and the effect on the victim, as they are often different. For example, theft of resources may be motivated to cause damage to the organization rather than to benefit the perpetrator. We concentrate on the majority of incidents caused by disgruntled technical employees that cause organizational damage by first interfering with IT systems and data.

3.2 The Essence of Insider IT Sabotage

CERT and SEI researchers analyzed 80 cases of insider IT sabotage in the United States between 1996 and 2007 [9]. There is enough of a pattern to decide useful controls to defeat likely threats and avoid damaging business efficiency by unproductive scattergun measures.

Insider attacks cannot be completely avoided, so we try to predict and avoid the likely ones, and strengthen the system to avoid a disastrous impact from successful ones. We ask the incident questions to determine the likely perpetrators, their intentions, actions and effects, which allows the determination of cost-effective protection offering high return on security investment (ROSI).

We investigate destructive insider attacks as an illustration of our methodology in practice. We illustrate some possible sabotage incidents using the iAssemble case study in table 1 with a more general table of damaging incidents elsewhere [10].

The goal of the destructive insider is usually the psychological satisfaction obtained from causing damage to the organization, or possibly other employees, motivated by a personal grudge for some perceived wrong. The aim is to destroy or harm physical assets such as buildings, equipment and computers; logical assets such as services, programs and data; or social capital such as organizational cohesion, reputation and financial health, or the psychological and physical wellbeing and health of employees.

These attacks typically interfere with the integrity and availability of system assets with the ultimate effect of interfering with the organization's abilities, activities and services. Therefore, sabotage is the means (how) using technical methods (with what) of achieving the ends of causing damage (what) to the organization (to what) that satisfies the motivation of a disgruntled employee (who) after some dispute (why) with the organization.

Logical insider sabotage is usually carried out by privileged IT staff (who) using unauthorized access with compromised accounts (with what). The active incident often originates outside the system (where) after the employee has left the organization (when), accounting for more than half of incidents [8]. There is also the possibility of an insider attack on an organization as a means (how) rather than the ends (what), such as attacks on critical infrastructure to cause wider damage to society (why) [9].

4 iAssemble Case Study

4.1 Background and Comments

The iAssemble incident is a fictional case study developed and used for training purposes by CERT and SEI [9] that is representative of the many real cases of insider sabotage from their research. We show a small number of possible sabotage attacks by Ian Archer, a disgruntled iAssemble employee, in table 1, which would be greatly extended in a realistic analysis [10]. We indicate some corresponding defences later, indicating defensive actions that may have been successful against each of these attack vectors to provide comprehensive defence-in-depth at all levels.

The iAssemble incident is summarized below with certain phrases highlighted to make a clear connection to our incident classification system. The summary provides a clear overview of the incident causes and effects. However, the response measures given are inadequate, because they are provided in hindsight with complete knowledge of the incident, whereas the victim needs to investigate all the likely actions of the perpetrator to rule out other malicious actions. The narrative is a slightly condensed version of the CERT description [11], except that the italics are added for emphasis and the text following the arrows are our comments.

'iAssemble sold computer systems directly to customers; building each system made-to-order at competitive prices. Ian Archer, the insider threat actor, had been with iAssemble since its founding and was the sole system administrator.

→ Answers the perpetrator question (who), and his power, knowledge and abilities (with what) over the system (to what), which can be abused for many purposes (why). His actions (how) can cause a great impact if he sabotages (what) computer production as the ultimate target (to what for the entire incident).

'Recent substantial company growth resulted in a change in culture, as well as new management who hired a *new lead system administrator*. This action triggered Archer's *disgruntlement*; he felt his hard work over the years was not appreciated.

→ Negative emotion (general why that provides context) is the first indicator of a potential sabotage incident shown in the left column in our incident table, as incident progression follows from left to right.

'The new administrator *restricted the privileges* of all iAssemble employees, including Archer.

→ Gives the perception of a sanction because of his previous free rein, together with the loss of autonomy and freedom (another general why for discontent).

'Archer vented his anger by openly *harassing individuals* and *stalling progress* on key projects.

→ Social-level behavioural indicator and technical sign respectively in the incident prologue rather than the active attack, but suggestive of serious discontent and indicate that malicious damage may follow.

'A performance improvement plan was instituted by Archer's new manager with *disciplinary actions* including written warnings, a temporary suspension, and reduction in his salary.

→ Sanctions often have the opposite of the desired effect of improving behaviour. The disgruntlement now turns into a positive intention (a specific why) to damage (what) a specific target (to what). The time (when) that damage is likely to occur is around termination by the pattern of sabotage incidents [9], and a system Archer controls as the location (where) as people typically attack systems they know well [8].

'Suspecting he would soon be fired, Archer created a *back door with system administrator privileges* on iAssemble's server for later access should his authorized access be disabled or his administrative privileges be revoked.

→ A typical move acting proactively to retain remote access (how and from where).

'Management's increased sense of risk of malicious activity led them to *ramp up audits of access control quality and access management*.

→ A necessary action to remove the ability to attack (how) once loyalty (a restraint on why) is lost.

'Unfortunately these measures were taken too late to prevent or detect Archer's backdoor installation.

→ Archer controlled the system previously because of lax oversight, so it is too late (when) after the situation has become critical to detect all backdoors and vulnerabilities into the system (where).

'When management fired Archer they disabled all known access paths. But unknown to management, a co-worker had *shared his password* with Archer to increase productivity for their project team.

→ A social-level breach of trust that retains access (with what) to the system.

'Archer used that password to log in remotely to the co-worker's machine on the night of his firing.

→ The action is the how, the location is the where and the timing is the when of the access stage. Both remote access (where in location) and an accessible account (where at the logical level) are needed to perpetrate an external logical attack. Using other accounts is a typical action of a malicious technical employee [8].

'Using the backdoor account he installed a logic bomb on the machinery server, set to detonate three months later.

→ Another typical method (with what) to avoid responsibility remote in time (when) and space (where) from the initial breach. The initial access is used to maintain subsequent indirect access using malware.

'The logic bomb deleted all files on the machinery and backup servers leaving the assembly lines frozen.

→ The logic bomb is the agent (who – software entities can be subjects), the target is the software files (to what) on all the servers (where) and the effect is the deletion (what). The assembly lines are the ultimate to what, the machinery at the physical level is the where, and stopping production is the ultimate what that meets Archer's intention (why) for the entire incident.

'An investigation revealed that access control policies and practices had eroded over time.

→ Access is the first active incident stage. There are two others stages of system misuse and causing a damaging effect, where independent controls can be placed to offer defence-in-depth. There should have been independent backups not available to Archer (where), which would have limited the ultimate effect (incident what). The inability of iAssemble to recover production rapidly had a major organizational effect from failing to supply their products leading to financial losses (incident what). We now move on to the repercussions in the epilogue stage.

The investigation led to the *arrest of Ian Archer*, and iAssemble's *share prices plummeted*.

→ The crime was hard to prove, because of inadequate evidence. The failure of iAssemble to provide its products and services adequately has knock-on financial effects (ultimate what) that were much worse than the direct losses (immediate what).

'Their *image in the market was blemished*.'

→ Often reputational damage (ultimate what) at the social level (where) is the most damaging effect.

4.2 Case Analysis

The iAssemble incident is shown as a path within a conceptual space in table 1 that implicitly shows the location and timing of incidents. The paths through the grid from left to right show incident progression through the various stages starting with access before using the target and finally causing the damaging effects. We can then consider proactive measures, protective barriers, monitoring processes and corrective actions in corresponding defensive tables (not shown) [12] to provide comprehensive and consistent defences at all layers to prevent, limit, detect and recover from Archer's malicious behaviour.

The top row in table 1 has cells for the perpetrator, motivation, ultimate effect and ultimate target that only have meaning at the social layer. The table omits or merges some columns whose content is clear. Archer is a disgruntled former employee that targets the organization for his perceived mistreatment, so the ultimate target is omitted and the attacker and motivation columns are merged to save space.

We attempt to explain each stage of access, use and effect carefully to avoid confusion. There will be at least one of each type of stage in a successful incident, but there could be more. Archer's incident involved multiple access stages, as he used remote access and a compromised account to plant a logic bomb to maintain indirect access.

We show the progression of the incident with arrows. The stages of access, use and effect are marked 1, 2 and 3 respectively, with multiple stages of the same type marked as a, b, c etcetera. In a more complex example, we would place the separate stages in their own tables rather than overlay them in a single table as here.

We focus on the main execution path of the iAssemble incident, but note possible offshoots that should be considered in a real incident, as the victim would be unsure about the sequence of events and would have to investigate all realistic possibilities. For example, Archer's remote access may have been discovered earlier, but the organization still needs to search for any other malicious actions, as in this case where he planted a logic bomb to detonate later.

Table 1. Some of Archer's possible incident paths

Perpetrator (who) and Motivation (why)	**Stage agent (who)**	**Reason (why)**	**Method (with what)**	**Action (how)**	**Target (to what)**	**Immediate Effect (what)**	**Ultimate effect (what)**
Disgruntled former employee gains psychological satisfaction from revenge for perceived mistreatment **Social 1a**	Archer. **1a** The targets unwittingly help to give unauthorized access to Archer **1b**	To gain system access	Social engineering ♣ **1a alternative path** *Social engineering using logical method* ↓	Persuade or trick target to act incorrectly by giving access, setting up accounts, or giving out passwords	Security guards, system administrators, colleagues ♦	Unauthorized physical, or logical access (via a compromised account)	Inability to produce computers → Failure to satisfy contracts → Financial losses → Reduced reputation, lost customers, lowered share price
Logical 1a part			Using email	Request password reset			
Logical 1b	Archer using his account	To gain hidden logical access after termination and avoid accountability	Misuse authorized authority using own account to issue commands	Logically authorized (but prohibited by policy at the social level) commands to set up a backdoor	Network access to system	Gain unauthorized remote logical access after termination	
Logical 1c	Archer using a compromised ♥ account and remote access	To install malware to maintain indirect system access and avoid accountability	Use of compromised account and remote access to issue commands	Unauthorized commands to install logic bomb	Operating system of computer holding production software	Installation of logic bomb *Backup loss*	
Logical 2/3	Logic bomb	To cause immediate damage to software → Ultimate goal to damage computer production	Privileged software misuses host system	Issue damaging commands to destroy files and software May also cover tracks by deleting log files	Software on production control system and backups → Production processes Log files ♠ ●	Unavailable production software → Lost computer production	
Physical	Archer *Physical attack goes from left to right cell to cell as logical attack*	Render critical assets unavailable → Stop production line	Misuse allowed access to physically interfere with equipment and resources before leaving, illegitimate access afterwards	Physical damage and destruction, theft, encryption	Software, backups, production computers and other essential equipment Logging devices	Damaged or unavailable systems and resources → Lost computer production	

The first stage of the iAssemble incident gains access using either authorized access, shown as a forward arrow from the perpetrator to use either logical (shown 1b) or physical access (arrow to the bottom row omitted), or backwards from the immediate effects of the first stage that gains unauthorized access. This occurred in the iAssemble case, shown by the backwards line marked ♦ from the immediate effect of acquiring the password to a colleague's account in the top row to its use at the beginning of the logical row to install the logic bomb.

The top row shows that Archer got a colleague to share their password, ostensibly to ease the performance of a legitimate task, but allowing subsequent access because passwords were not changed when he left. He could also launch a social engineering attack on a security guard to trick them into giving unauthorized physical access after termination, or on a system administrator by masquerading as another employee to get their password reset, which could be shown as alternative paths that terminate with the same immediate effect at the social level when unauthorized access is achieved. Sending email requesting a password to be reset using another employee's compromised email account is a logical action exploiting the inadequate authentication provided by email, shown by the detour to the logical level marked ♣, which also has a path that eventually moves back up to the targeted system administrator at the social-level.

The other access acquired by Archer was to ensure he had remote access, needed for a logical attack from outside after termination. He would otherwise need physical access to permit local logging on to the system, by getting past the security guards. Incidents using physical access to perform logical attacks show the complex interaction between the levels, showing the need for a systematic model.

Gaining remote access is shown within its own row at the logical level marked 1b by misusing authorized access using his own account to install a hidden backdoor with the immediate effect of allowing logical access after termination and helping to avoid accountability. A similar act (not shown) at the physical level, would be to install a hidden ADSL modem or a wireless access point before leaving.

The access stage is not a complete incident, as breaches of security mechanisms do not usually directly interfere directly with organizational goals, which is shown by the arrows not reaching the ultimate effects column, but instead passing down to the start of a subsequent attack stage using the acquired access. The arrows cross the levels of the table showing informally the passage through a logical or physical boundary to gain lower-level access to the system. We have already seen level crossing with the attempt to get a password reset using email.

Archer then launched a logical attack using his colleague's compromised account along with the remote access to install a logic bomb to delete the production software and all backups. The table distinguishes between unauthorized access that requires a traversal of the table from left to right first such as 1a to gain access via a compromised account, and authorized access marked 1b that passes forward without requiring an initial access violation stage to set up remote access using his account.

The table also demonstrates how the access gained in the first stage is used to gain further access. The installation of the logic bomb requires both the remote access and account compromise at point ♥.

The installation of the logic bomb is the launch pad for the use stage, where the detonation of the bomb to destroy the production software is shown in the lowest

logical row marked 2. The effect stage includes the subsequent use of the targeted resources by the perpetrator and their escape. The effect of a sabotage incident is often coincident with the use stage as here, because the execution of the logic bomb has the immediate effect of destroying the production software, and so consequentially, stage 2 is also marked 3 as shown.

This resulted in the lost production of computers, which was exacerbated by the failure to provide independent protection for the backups. The diagram helps conceptually to provide defence-in-depth by indicating where independent controls can be placed to obstruct the use and effect paths after unauthorized access.

Clearly, there are many other paths that Archer could have followed, which should be investigated and whose discovery is aided by our diagrammatic representation. For example, physical access could be obtained by tricking a security guard, which would be shown by a backwards arrow from gaining unauthorized access in the top row going to the beginning of the physical row analogous to ♦, that could then be used to steal or damage physical resources along a path in the bottom row. Note the unconnected arrow marked ♠ in the bottom row, where physical destruction or removal of the software container has a logical effect. The rest of the initial horizontal physical path to access the software is not shown to avoid overloading the diagram.

The impact at lower levels from damage to resources and services is eventually transformed into organizational social-level difficulties. As mentioned before, the access stages do not cause an ultimate effect, except to cause disruption to find and repair the exploited weaknesses. Sabotage has the immediate effect of compromising the availability and integrity of the targeted resources (means) usually at lower layers, with the ultimate effect of damaging the organization's ability to carry out its normal business activities (ends). This is shown as an arrow that moves from a lower level to the social level in the last column showing the ultimate effect.

The software controlling computer production was damaged (means) at the logical level with the ultimate effect of damaging the organization's ability to produce and sell computers (ends), which caused financial damage and had knock-on effects on reputation and share price (ultimate ends) that met Archer's motivation (why).

Employees are in a good position to target critical system weaknesses such as essential components that are difficult to repair or replace. In addition, they can interfere with the recovery mechanisms, which can be included in the incident table as additional steps that disrupt the defence. Archer achieved his ultimate objectives, because of the single point of failure of the production software that was centralized in one location, and his destruction of the backups that were all logically accessible by the logic bomb. This stopped effective recovery causing a significant outage, rather than straightforward and speedy restoration from backups, which could have led to permanent loss of crucial business assets and even organizational failure.

The table shows the destruction of backup data marked •. The effect is shown by a special arrow to the arrow between the logical and social levels strengthening the damage to the organization, as destroying backups is a latent weakness that intensifies the incident effects.

Physical attacks in particular are often overlooked and very hard to stop against determined insiders. The upward arrow marked ♠ shows the reliance of software and other logical resources on their physical storage containers.

4.3 Defensive Response

We respond to the incident questions in the order shown in the incident table to analyze the separate aspects of protection.

Who? The saboteurs were chiefly male and held technical IT positions with the majority having system administrator or other privileged access. This means they have a high level of authority, access and knowledge of computer systems making them very dangerous adversaries. Rather surprisingly, the majority of the insiders who committed IT sabotage were former employees at the time of the incident [8].

Why? The *why* is answered by the personality of the perpetrator and more specifically by the negative workplace events. Most perpetrators were disgruntled, acting out of revenge for some negative precipitating workplace event such as termination, disputes, new supervisors, transfers or demotions, or poor remuneration.

The organization should use both 'carrot and stick' to persuade employees to behave correctly and deter them from acting maliciously. Persuasion is about influencing people with positive motivation and incentives to do the right thing (increasing the positive *why*). Deterrence impedes people from doing the wrong thing, by making the consequences worse than the gains (increasing the negative *why*).

When? The majority of saboteurs were former employees acting after leaving the organization, although they often took steps before to retain access by creating backdoor accounts, planting logic bombs, or amplifying the attack by destroying or removing backups beforehand. More than half attacked outside of normal working hours using remote access [8].

Where? More than half of saboteurs attacked from outside using remote access, and targeted a system to which they had authorized access presently or in the past. These incidents involve logical systems in two different conceptual locations, and so we need to consider the security of both endpoints along with the connecting pathway. IT saboteurs generally attack logical systems to cause the biggest possible disruption, but physical systems also need adequate protection. We need to limit access at all levels by providing a comprehensive attack surface involving both physical and logical protection.

With What? The perpetrator uses methods, tools and other resources in an incorrect or illegitimate way. One protective measure is to reduce the perpetrator's authority over and access to resources. Archer's complete access to the production system meant that he could deploy multiple methods against it. The defence may also reduce system and resource functionality to avoid abuse. Remote access would be ineffective if the production systems were isolated from the Internet. Similar controls on use of physical resources are needed too, covered by our multilevel model.

How? The majority of IT saboteurs did not have authorized access at the time of the incident. They generally used other accounts including other employees' accounts, shared accounts such as administrator, or they set up new ones. A third used

sophisticated technical means such as logic bombs. Most of the insiders took steps to conceal their actions by modifying or deleting the logs [8]. All of these activities are relevant to the iAssemble incident.

At the social level, there should be effective procedures to stop Archer from gaining unauthorized access through physical admission, setting up new accounts, or password sharing. At the logical level, independent monitoring and oversight may have detected the development of the logic bomb as it was tested extensively before detonation. Network access controls and intrusion detection may have detected and blocked the remote access used to install the logic bomb. Physical access to equipment should be limited, as it may cause the same devastating impact as logical attack. This involves protection of the hardware containing the production software, and having independently protected redundant resources, such as offsite backups.

To What? The target is usually a system the saboteur knows well, so they knew weaknesses to exploit to cause maximum disruption. They may interfere with access controls to gain access that is then misused during the use stage. In other words, the target (to what) of one stage is often used as the method (with what) of a later stage. Archer used the colleague's compromised account, and his legitimate access to gain remote access, using both at point ♥ to install the logic bomb (all stages of increasing access), which was then used as the means to damage the software that was the immediate incident target.

The most important design principle is to avoid a single point of failure at all levels to assets whose loss would have a significant impact. Archer developed the production software, and he should never have been given complete control.

What (Immediate Effects)? The immediate effect of sabotage is to cause damage to networks, systems, software and data. The loss of resources will have a lasting effect on the organization if they are essential, and cannot be recovered, rebuilt or repurchased.

What (Ultimate Effects)? There were ultimate effects at the social level in two-thirds of sabotage cases [8] with an organizational impact from the inability to conduct business when the system or network is unavailable, or to produce products and services because of damaged or destroyed systems. Other negative consequences come from loss of reputation, which occurred at iAssemble from lost production and resulting failure to satisfy contracts. They needed to have effective disaster recovery and business continuity processes in place, but Archer was the software creator and thus a single point of failure.

5 Conclusions and Further Work

The insider threat poses a significant and increasing problem as employees' loyalties are often questionable and organizational boundaries become blurred. Systematic defence is required as no single method can protect against employees with legitimate access to organizational resources. We proposed an architectural three-layer security model to analyze the insider threat systematically, and broadened Howard and

Longstaff's classification [6] to all three levels. We extended the six questions from Zachman's framework to eight to pose questions about each aspect of incidents to enable comprehensive analysis.

We introduced the idea of stages, where the active incident is divided into three stages of access, use and exit as a methodical way to analyze incidents. This enables a systematic determination of possible defensive measures at all levels and locations to limit access, constrain use of the target and reduce the impact of successful attacks.

The systematic method using tables that we used to analyze destructive attacks can clearly be extended to the other main classes of insider attack from fraud and theft. However, the tabular form is unwieldy to analyze complex incidents and better visualization techniques are required, which is the focus of ongoing work. Finally, we are investigating the link to system dynamics [11], especially concerning proactive incident prevention, rather than in the analysis of extant incidents as here.

References

1. Blackwell, C.: A Security Ontology for Incident Analysis. In: 6th Cyber Security and Information Intelligence Research Workshop. ACM press (2010)
2. Tanenbaum, A.S.: Computer Networks, 4th edn. Prentice-Hall (2003)
3. Zachman, J.: A framework for information systems architecture. IBM Systems Journal 26(3) (1987)
4. Department of Justice: Digital Forensics Analysis Methodology. Department of Justice (2007), http://www.justice.gov/criminal/cybercrime/forensics_chart.pdf
5. Howard, J.D.: An Analysis of Security Incidents on the Internet, 1989 – 1995, PhD thesis. Carnegie-Mellon University (1997), http://www.cert.org/research/JHThesis
6. Howard, J.D., Longstaff, T.A.: A common language for computer security incidents. Sandia National Laboratories (1998), http://www.cert.org/research/taxonomy_988667.pdf
7. Cappelli, D.M., Moore, A., Shimeall, T.J., Trzeciak, R.: Common sense guide to prevention and detection of insider threats, version 2.1., CERT (2006), http://www.cert.org/insider_threat
8. Cappelli, D.M., Moore, A., Shimeall, T.J., Trzeciak, R.: Common sense guide to prevention and detection of insider threats, version 3.1., CERT (2009), http://www.cert.org/archive/pdf/CSG-V3.pdf
9. Moore, A.P., Cappelli, D.M., Trzeciak, R.F.: The "Big Picture" of Insider IT Sabotage Across US Critical Infrastructures. TECHNICAL REPORT CMU/SEI-2008-TR-009, Software Engineering Institute, Carnegie Mellon University (2008)
10. Blackwell, C.: A Framework for Investigative Questioning in Incident Analysis and Response. In: 7th Annual IFIP WG 11.9 International Conference on Digital Forensics. Advances in Digital Forensics VII. Springer (2011)
11. Cappelli, D.M., Desai, A.G., Moore, A.P., Shimeall, T.J., Weaver, E.A., Willke, B.J.: Management and Education of the Risk of Insider Threat (MERIT): Mitigating the Risk of Sabotage to Employers' Information, Systems, or Networks. TECHNICAL NOTE CMU/SEI-2006-TN-041, Software Engineering Institute, Carnegie Mellon University (2007)
12. Blackwell, C.: The insider threat: Combating the enemy within. IT Governance (2009)