THE INSIDER THREAT MINIMIZATION AND MITIGATION FRAMEWORK

Ghassan (Gus) Jabbour[1] and Jason J. Jabbour[2]
[1]*Principled Innovation Institute, Fredericksburg, VA, USA*
[2]*University of Virginia, Charlottesville, VA, USA*

ABSTRACT

Countering the insider threat is a difficult and daunting task. Organizations concerned with the problem usually train their employees on security-related matters, rules of behavior policies, and the consequences of committing criminal activities. More technically-oriented solutions include enhanced credentialing and access control, and the use of monitoring tools that provide insight into the health and status of systems. This paper addresses the deficiency of widely-used monitoring tools and strategies. It proposes a solution that equips a system with innate self-defense mechanisms that relieve the system from having to rely on human intervention. The paper introduces the Insider Threat Minimization and Mitigation Framework. The framework equips systems with self-defense mechanisms such that a system can instantaneously respond to potential threats and defend itself against users who have unfettered access to it. The framework employs the autonomous demotion of power users' access privileges based on analysis and evaluation of the user's risk level. The paper presents the details of the proposed framework and simulates its effectiveness within a data center environment of mission-critical systems.

KEYWORDS

Insider Threat, Autonomous Computing, Self-Protection

1. INTRODUCTION

The Insider Threat continues to be a major risk to the security and financial stability of businesses. The most prevalent types of insider attacks are those associated with the theft of intellectual property, sabotage, fraud, and espionage. Countering such threats is a significant challenge. Today, there are a plethora of security software that offer system monitoring, incident management, and threat investigation capabilities. Usually, these are services provided by cloud service providers or vendors managing a data center. These tools constantly connect to and check system resources then generate alerts, notifications, and reports. The main use of such tools is to provide insight into the health and current state of a system. They do so by tracking specific system metrics in real time and generate alerts when certain readings exceed or fall below preassigned thresholds. This paper introduces the Insider Threat Minimization and Mitigation (ITMAM) Framework that goes beyond the usual monitoring of systems to enabling such systems with self-protection mechanisms. What distinguishes the ITMAM framework is that it inherently equips the system with defense capabilities allowing it to protect itself from potential insider attacks. Based on autonomous computing concepts, the ITMAM framework enables the system to intelligently deploy various levels of defenses in response to various potential threats without having to rely on human intervention. The use of autonomous computing capabilities provides a superior advantage over approaches that require human intervention. Conventional software monitoring tools are heavily deployed and used at data centers across the world. Companies rely on such tools to monitor the health of their systems, and to enforce and comply with recommended or federally mandated security requirements. Such requirements are either suggested by standards organizations such as NIST or the CERT Division at the Software Engineering Institute at Carnegie Mellon University, or they are federally mandated by laws such as the Federal Information Security Management Act (FISMA) of 2002. There is no doubt that such tools have been effective at detecting suspicious activities and malicious acts. They are also effective at alerting system owners of potential system failures. However, such tools do not address the crux of the issue that lies at the heart of the insider threat problem. As long as power users have unfettered access to the system they manage, they can inflict tremendous damage to it before they are caught. System, database, or network

administrators have, by design, unfettered access to their systems. Therefore, they have the power to alter or damage the system instantaneously. Thus, as long as the defense mechanism lies outside the system that is being protected, there is no guarantee that the system will be protected. Autonomous computing equips the system itself with self-* capabilities (such as self-protecting, self-optimizing, self-managing, etc.) that move the control and command capability into the core components of the system itself. By doing so, the system can protect itself from actions that it deems potentially malicious or intended to damage it. Clearly, that comes with a certain level of restriction and limitation imposed on power users. However, those restrictions are specific to only certain system functions that are usually intrusive in nature. The idea is to create a balance between the act of self-defense and allowing administrators to perform their duties. We expect that if autonomic computing is properly designed and applied, such a balance can be struck. Monitoring tools are unable to prevent malicious actions by a power user who has unfettered access to the system. This deficiency is addressed by the ITMAM framework as it empowers a system to depend on itself rather than wait on delayed, or too late, human intervention. This paper presents the ITMAM framework and describes its various components. Also, it demonstrates its benefits by exercising its capability in a simulated data center environment.

2. LITERATURE REVIEW

The number of attacks committed by insiders, and the consequential damage, has been significantly rising in the past few years. In June of 2018, a former employee of Tesla allegedly “hacked the company's confidential and trade secret information and transferred that information to third parties" (Schwartz, 2018). In November of the same year, the hotel chain Marriott was hit by an insider attack where the records of 500 million customers had been compromised by an unauthorized party (BBC News, 2018). However, the insider attack committed against SunTrust Bank in 2018 was the costliest. The company, consequently, disclosed that one of its employees worked with outside criminals to gain access to, and compromise, the accounts of around 1.5 million clients (Weise, 2018). Moreover, the number of insider threat incidents and their related cost keeps rising year after year. According to Verizon’s 2019 Data Breach Investigations Report, 34% of breaches that took place in 2018 were caused by insiders (Verizon, 2019). Notably, what elevates the insider threat problem to a much higher risk level is that a significant percentage of insider attacks are committed by power users. According to the 2019 Insider Threat Report produced by Cybersecurity Insiders, privileged IT users pose the biggest insider security risk to organizations (62% of total attacks) (Schulze, 2019). As such, companies cannot trust them with unfettered access to their systems. A system must be able to defend itself against power users. Any defense mechanism that leaves power users with unfettered access to the system is ineffective and weak. This validates the need for a framework that delegates the system protection task to the system itself. In order for that to work, the self-protection capability must be totally integrated into, and inseparable from the system that is being protected. Also, that capability must not be accessible by system administrators. There are many system monitoring tools on the market today that are being advertised as system protection and defense tools. However, almost all of these tools focus on monitoring the systems to allow administrators to manage their IT infrastructure efficiently, improve its uptime, respond to threshold alerts, and reduce cost. Many of these systems rely on analyzing system-generated data, identify vulnerabilities, identify potential threats, and report the findings to the data center leadership for their review and action (Ikany & Jazri, 2019), (Mavroeidis et al., 2018), (Ali et al.,2008). Others tackle the threat detection task by viewing it through different layers of the system namely the application layer, support layer, network layer, and perceptual layer (Kim et al., 2020). Some use situation-awareness technologies to detect an insider threat (Buford et al., 2008). Yet, other approaches focus on behavioral differences and social context variables using multidimensional classification systems that consider the severity (S), intentionality (I), type of employee norm violation (EV), and ethicality (E) of the incident (Schoenherr & Thomson, 2020). Lastly, some researchers have focused on developing a hierarchy-mapping-based insider threat model that described the kernel of threat detection, using sense and prediction to detect insider threats (Zhang et al., 2009). While such approaches have their merits, they are not effective in neutralizing or preventing an attack by an insider who has unfettered access to the system. Current monitoring tools leave it up to the system administrator to take action as discussed in Ali et al (2008), Paci et al. (2013), and Beena, Humayoon Kabir (2019). Such systems become obsolete and ineffective if the power user/administrator is the attacker. This is because a power user can perform an attack instantly thus rendering the alert and notification approach totally obsolete. This is supported by a study presented in Claycomb et al.

(2013), where the authors concluded that "the acts of IT sabotage often happened very close, if not in conjunction with, the moment of cyber damage to the organization". The same study also revealed that "90% of the cases studied will have an observable event indicating potential malicious activity prior to the moment of actual damage". This also validates the ITMAM framework, which allows the system to defend itself from within. This is supported in Chen & Lambright (2016), which argues that a self-protecting system "will perform reliably, trustworthily and resiliently even if it is compromised by cyberattacks". However, the ITMAM framework advances this notion by showing that a self-protecting system will not only perform resiliently but is also capable of completely preventing the occurrence of an insider attack even when attempted by the administrator of the said system.

3. THE ITMAM FRAMEWORK

The critical distinction between the ITMAM approach and that of other monitoring and threat detection tools is that the ITMAM capability lives inside of, and inseparable from, the system that is being protected. It simply delegates the task of protecting the system to the system itself. It does not require human intervention to invoke its defense capabilities to reject the action of a power user if deemed malicious. Also, the ITMAM self-defense capability is not accessible by the administrator. Technical details of how a defense mechanism is completely embedded into, and inseparable, from the system that is being protected are explained in Jabbour & Menasce (2009a) and Jabbour & Menasce (2009b). This paper argues that for any monitoring, threat detection, and mitigation methodology to be successful, it must be totally embedded and inseparable from, the system that is being protected. Moreover, the paper presents a detailed illustration of how the ITMAM framework works in equipping a system with self-protection capabilities. Expecting external monitoring and threat mitigation tools to fully protect the system is a weak and vulnerable protection mechanism. The reason is that such monitoring tools can be easily interrupted by the administrator. Alternatively, defense solutions based on the ITMAM framework are totally embedded into the system being protected and are inseparable from it. Once the solution is embedded into the system itself, the system will then be able to detect threats and protect itself without reliance on humans. It does so by predicting and rejecting unacceptable actions before they are even submitted. As discussed earlier, the ITMAM framework is designed for a data center environment where mission-critical systems reside. Such systems could potentially be the target of malicious attacks by insiders who wish to cause serious financial, social, or even national security damage. The ITMAM solution works by constantly collecting and analyzing user-related data to produce actionable information that give the system itself the power to autonomously detect and mitigate a potential threat before it happens. Insider attacks are usually committed by company employees who have elevated access privileges to mission-critical systems. However, elevated access privileges are intentionally given to employees to properly perform their duties. So, the consequences of such employees becoming malicious and committing criminal activities are significant and very costly. This paper argues that equipping the system with the ability to protect itself is much more effective. The ITMAM framework has two main components/modules that contribute to detecting, minimizing, or totally mitigating a potential insider attack. These are:

1) Collection, Aggregation and Synthesis of Employee-Related Actionable Information (CASAI): the CASAI component of the framework is tasked with collecting data from multiple sources to form a comprehensive and meaningful record of the risk level or threat status of an employee. The data sources are the company's various information systems as well as publicly available systems. Data is collected to enhance the understating of a potential threat level associated with an employee. The CASAI component only collects and stores data associated with positive hits or suspicious data that could lead to threat aversion. Once formed, an employee-related CASAI dataset is sent to the ADEMPT component for the investigation and mitigation of a potential threat.

2) Autonomic Detection, Evaluation, and Mitigation of Potential Threats (ADEMPT): the ADEMPT component equips the system with self-protection capabilities to counter any malicious action by power users. Equipped with built-in intelligent predictive analysis capability, strict business rules, and detailed security settings, the system invokes its self-protecting mechanism when needed to protect itself. Embedded self-protection mechanisms allow the system to detect a potential threat, evaluate possible countermeasures, assess unintended consequences, and take the appropriate steps to minimize or totally mitigate a threat. It performs assessments and predictions based on real-time access to employee and data center data.

Data is collected in real time from the various sources and assimilated to form the basis for predictive analysis of a user's risk level. The data sources are Data Center Point of Entry, Employee Performance

Appraisal Systems, Employee Grievance Management Systems, Employee's Corporate Email Traffic, Employee's Text Message using Corporate or Government Furnished Equipment (GFE), Internal Web Traffic, Public Social Media Sites, Financial Systems or Records, Law Enforcement Public Records, Network Traffic and Packets, System Event Log Files, Biometric Data Systems, and Data Center Physical Environment Sources including motion detectors, microphones, smart thermometers, and hygrometers. Existing tools also use such data; however, they use it in a reactive manner. They make the data available to managers or administrators to react to it. Instead, the proposed framework delegates that authority to the system itself. The rest of the paper presents the ITMAM Architecture and demonstrates its effectiveness in detecting and mitigating an insider attack.

4. THE ITMAM ARCHITECTURE

The ITMAM architecture is designed to empower a system with self-protection capability. Therefore, the system must be able to capture data from multiple sources and learn its environment and the actions of its power users. Based on that data, and in conjunction with predefined system security settings, the system is able to decide whether to accept or reject an action submitted by a power user. As discussed earlier, the components of the framework must be integrated into the system that is being protected. For example, in a Database Management System (DBMS), the components of the framework are codified into database objects at the system level, and are not available to the database administrator, but will control the actions of the database administrator to protect the system as explained in Jabbour & Menasce (2009a) and Jabbour & Menasce (2009b). A full depiction of the ITMAM architecture is shown in Figure 1.

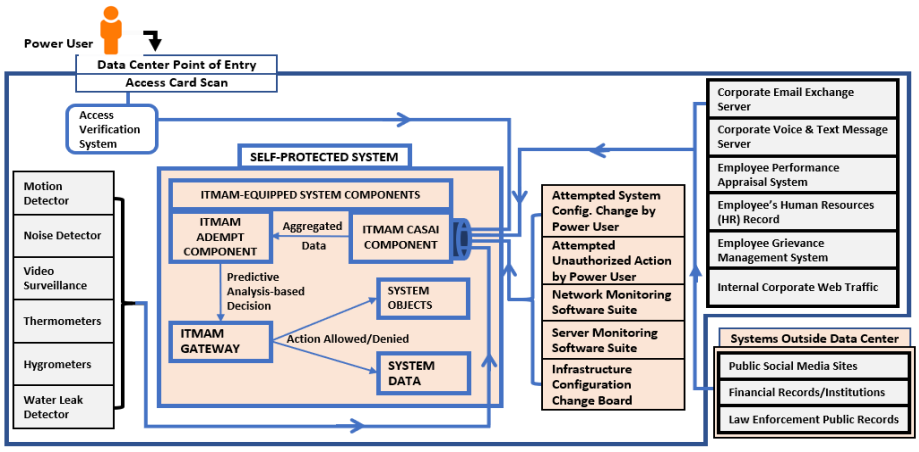

Figure 1. Architecture of the ITMAM framework

The architecture shows an ITMAM-equipped system residing inside a data center. The ITMAM components of the protected system gather data from internal systems as well as systems outside the data center for analysis. The protected system then performs predictive analysis to determine if an action is a threat. In response, the system acts instantaneously to neutralize or totally block an attack before it takes place. Systems on the left of the diagram are used to capture information on the state of the physical environment of the data center. That information is then combined with data from the other systems to form a complete understanding of what may be taking place inside the room. Systems on the right of the diagram are divided into two types. Some belong to the company and are hosted within the data center. Others are publicly available systems that are hosted somewhere outside the data center. It is important to note that the self-protected system could potentially be all of the systems within the data center. Figure 2 presents the Data Flow Diagram (DFD) of the ITMAM framework. Data collected from sensor devices, such as smart thermostats and noise and motion detectors, are fed into the CASAI module on an ongoing basis. However, data from the other systems (such as an employee's HR record) are requested in real-time as soon as the employee's identification information is captured at the data center's point of entry. At the same time, data from social media sites and other public websites are searched using bots.

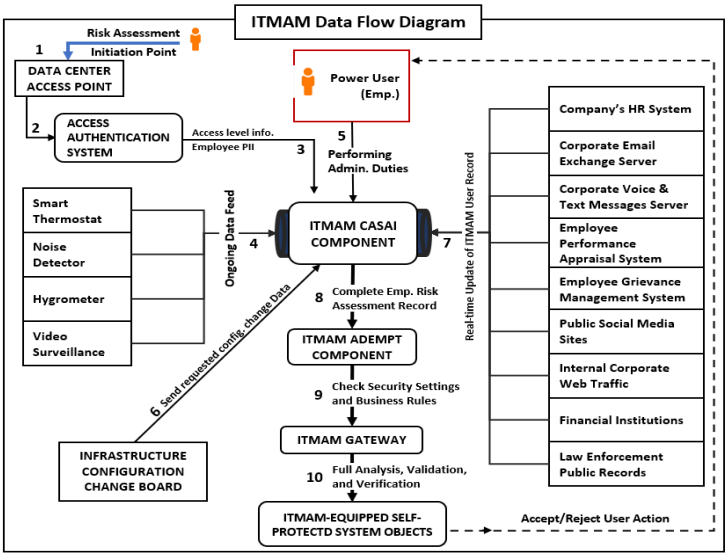

Figure 2. The ITMAM Data Flow Diagram

The data flow process is initiated when a power user accesses the data center at the point of entry as shown in step 1 of the DFD. At that point, an initial shell record of the user's data is created in the CASAI database. The employee's Personally Identifiable Information (PII) as well as data center access card authentication information are captured at the point of entry and sent to CASAI as shown in step 3. As soon as the user logs into the protected system (network, server, etc.) as depicted in step 5, the CASAI module requests and receives complimentary data from the other systems as shown in steps 6 and 7. The totality of the data received so far form the basis of the risk assessment in response to a power user submitting an action to the system (step 5). In this context, an action is any intrusive transaction that a user submits to the system such as a system configuration change, deleting system objects, injecting code, or installing or uninstalling unapproved software. The CASAI module collects the data, aggregates it with user actions, and sends it to the ADEMPT component for predictive analysis and decision making (step 8). Using the ITMAM GATEWAY, the ADEMPT component checks the predefined security settings and business rules stored within the system. It then uses that information to inform its analysis and decision-making process (steps 9 and 10). Depending on the result of the analysis of the complete user record, the self-protected system returns a decision to the user either accepting or rejecting the action.

5. FRAMEWORK SIMULATION AND VALIATION

To test the validity of the ITMAM framework, we simulate a data center environment where user-related data is collected, analyzed, and used to produce outcomes. The ITMAM interface allows the system owner to setup predefined security settings at the initial implementation and instantiation of the solution. Changes can later be made while the solution is already in operation, however, that change process follows a very strict and controlled path, where any change has to be approved by 3 or more members of the leadership team as discussed in Jabbour & Menasce (2009). For the purpose of this simulation, we chose 30 system security settings that, in normal setups, can be manipulated by a power user/system administrator at will.

Table 1. User actions and their corresponding threat level

	ACTION	Action Severity		ACTION	Action Severity
1	Install Malware on Network	H	16	Disable or Unistall Antivirus Software on Web Servers	H
2	Change password expiration period from 30 days to 45 days	L	17	Modify Domain Name System (add malicious domains)	H
3	Change password expiration period from 45 days to 30 days	L	18	Allow use of Portable Devices (USB Sticks, SD Cards, etc.)	H
4	Ignore Outdated or Unpatched Software when Prompted	M	19	Stop Data Backup Process	M
5	Disable Network Monitoring tools	H	20	Stop Transferring Backup Data to Secure Offsite Location	M
6	Disable Application Monitoring tools	H	21	Disable/Remove Network Encryption	H
7	Disable Database Monitoring tools	H	22	Use Credentials of a Different Power User	H
8	Upgrade Application Software	M	23	Disable Multi-factor Authentication	M
9	Update Security Certificates	M	24	Disable Password Expiration	M
10	Make Unauthorized/Unapproved Change of Firewall Config.	H	25	Install Approved Remote Desktop Software	L
11	Make Authorized/Approved Change of Firewall Config.	L	26	Install Unapproved Remote Desktop Software	M
12	Install Authorized & Approved Software on the Network	L	27	Enable Automatic Run/Open of Safe Files in Web Browsers	M
13	Install Authorized/Approved Software on Virtual Machines	L	28	Set Failed Login Attempt to Unlimited	M
14	Add Unauthorized/Unapproved Devices/Nodes to Network	H	29	Stop Database Audit Process	M
15	Disable or Uninstall Network Antivirus Software	H	30	Change/Elevate User Account Privileges	M
	Key: H = High M = Medium L = Low				

Most of the settings we selected are considered critical to the credibility, integrity, and availability of an information or network system. We also assigned a threat level to each parameter depending on the damage that the change of the security setting may cause if allowed. Those are presented in Table 1. The main threat levels we used for the simulation are Low, Medium, High, and Severe.

Table 2. Data elements of a user record

	USER RECORD DATA ELEMENTS	Description	Weighted Threat Level (Pts.)
1	USER_HR_FLAG	Background Check Red Flag, Unreported Job History, Employment-related Convictions, Discrepancy of experience vs. actual experience	30
2	USER_SENORITY_EXP	Unexperienced: ≤ 1Yr Mid Level Exp.: 1-5Yrs Senior Level: ≥ 5Yrs	1 Yr = 5 1 - 5 Yrs. = 15 > 5 Yrs. = 20
3	MISUSE_CORP_EMAIL	Comparison of email words to Predefined Ontology Repository	25
4	MISUSE_CORP_TXT	Comparison of text messages to Predefined Ontology Repository	25
5	MISUSE_CORP_VOICEMAIL	Comparison of recoded voice messages to Predefined Ontology	25
6	EMP_LAST_PREF_RATING	The employee's annual perforamnce rating	UNACCEPTABLE = 30 FAIR = 10 GOOD = 0 EXCELLENT = 0
7	EMP_GRIEF_CASE_6MO	Whether employee filed a grievance case in the past 6 months	20
8	MISUSE_CORP_WEB	Comparing visited sites against a repository of unfavorable sites to visit	25
9	EMP_FACEBOOK_POSTINGS	Whether employee posted derogatory words/phrases against company	15
10	EMP_TWITTER_POSTINGS	Whether employee posted derogatory words/phrases against company	15
11	SOCIAL_MEDIA_THREAT	Whether employee posted threat against specific employee(s)	YES/NO FLAG
12	CORP_ICCB_ACTION_APPROVE	If system change approved by Infrastructure Configuration Change	YES = 0 NO = 30
13	EMP_PUBLAW_VIOL_RECORD	Whether a criminal action has been recorded against the person	20
14	EMP_FINANCE_HARDSHIP	Bad credit or other negative financial information	10
15	RANDOM_SYS_FAILURE	Malfunctioning of the system due to failure not related to an attack	YES = 0 NO = 5
16	ROOM_TEMP	Data Center room temperature should not go above 82°F	≥ 82°F = 10
17	ROOM_NOISE_LVL	Normal range 40 - 70 dB but above 85 may be concerning	≥ 85dB = 10
18	DETECTED_WATER_LEAK	Whether the data center is experiencing water leak or flood	YES = 5 NO = 0

However, when combined with other data values collected from the various systems, a threat level can be demoted, elevated, or kept the same. For example, if elevated, a High threat level becomes Severe, in which case the self-protected system would totally lock the user out and promptly reject any submitted transaction. If changed, a threat level of Medium becomes either High, Low, Severe, or left unchanged. Next, we present in Table 2 the data elements that the CASAI module requests and receives from the various data sources that it interfaces with as shown in the data flow diagram. We assign a weighted value to each data element according to its relevance and the degree of impact it has on the decision of accepting or rejecting a user's action. Each data element is assigned a weighted score ranging from 1 to 100. The higher the score, the higher the risk. Depending on whether the overall weighted risk reaches or exceeds the allowable threshold of the ITMAM model, the threat level is elevated, demoted, or left unchanged. In response, the self-protected system invokes the appropriate outcome to deal with each specific threat. Table 2 shows 18 data elements that partially make up the user record that the ADEMPT module analyzes and investigates to determine the proper outcome. The table also includes the points associated with each data element. To cover the various security aspects of a data center, we have divided the threat assessment criteria into four categories. Those are shown in Table 2 as follows: Potential Insider Attack & Malicious Activity represented by data elements 1 – 14, Random System Failure or System Malfunction represented by number 15, and Physical Data Center Security represented by 16 – 18.

6. SIMULATION RESULTS AND FINDINGS

The simulation of the ITMAM framework utilizes a dataset of 150 records. The dataset is synthesized using realistic data center experiences of the relative occurrences of actions by system administrators. Also, the dataset is enriched with the initial system security settings and the outcomes associated with each action if allowed to be executed. Realistic and appropriate weights were assigned to each threat scenario. The simulation algorithm was designed and built to simulate the potential behavior of a power user from the moment he/she enters the data center. The algorithm generates one data record each time a user enters the data center. It then analyzes the complete user record to determine the risk level associated with the user. As a result, the simulation shows the outcome that the self-protected system undertakes to protect itself. The simulation algorithm also records the events, then uses that data to predict the next action of a power user. In this section of the paper, we present four instances from the simulated data and show how the system reacted to each one. In two cases, the self-protected system rejected the action and protected itself from the consequences. In the other two cases, the self-protected system allowed the action based on the overall analysis of the situation. Below are the four different scenarios:

```
Capture Employee ID: 8528
Employee profile found: [Name: Alisha Holland, HR Flag: False, Years Exp: 2, Perf. Rating: 4]
Capture threat information: Misuse Corp Email, Misuse Corp Text, Misuse Corp Voicemail,
Misuse Corp Web, Grief Case: 0,0,0,0,1
Misuse Facebook, Misuse Twitter, Direct Social Media Threat: 0,0,1
ICCB Action Approved, Law Violation, Financial Hardship, System Failure: 0,0,1,0
Room Temp, Room Noise, Detected Water Leak, Action id: 72,47,0,14
Action: Add Unauthorized/Unapproved Devices/Nodes to Network
Assessing threat severity...
[Weighted Threat: 80] [Default Severity: H] [Change Made: Increased by 1 Level] - [New Severity: S]
Outcome 1: USER REQUEST REJECTED
Outcome 2: USER ACCOUNT LOCKED ON ALL SYSTEMS. ENTRY TO DATA CENTER DISABLED
Outcome 3: Attn. Power User: Your account has been locked. Consult supervisor for next steps
Outcome 4: Alert sent to sec. team mobile device 3 times in 5 min and repeated in 5 min
Outcome 5: Urgent email sent to sup. & sec. team: Emp. Alisha Holland barred from
accessing all DC systems & physical access to DC revoked. Refer to ITMAM for incident details.
Outcome 6: Incident details have been stored in ITMAM for analytics & data mining purposes
```

Figure 3. Threat level elevated from High to Severe

Figure 3 shows a case where the threat level was elevated from High to Severe based on the insider's threat level. The action attempted by the power user was to add unauthorized devices/nodes to the network. A variable of the system algorithm, which stores data about a threat made by the user against a specific employee of the company was flagged. In this scenario, the initial threat level of the action was High. Based on the combined threat factors, the level was immediately elevated to Severe. Figure 4 presents a case where a Medium threat level has been elevated to High based on the context of all the data collected about the user.

```
Capture Employee ID: 2213
Employee profile found: [Name: Russell Morrison, HR Flag: True, Years Exp: 4, Perf. Rating: 1]
Capture threat information: Misuse Corp Email, Misuse Corp Text, Misuse Corp Voicemail,
Misuse Corp Web, Grief Case: 0,0,0,0,0
Misuse Facebook, Misuse Twitter, Direct Social Media Threat: 0,1,0,
ICCB Action Approved, Law Violation, Financial Hardship, System Failure: 0,0,0,0
Room Temp, Room Noise, Detected Water Leak, Action id: 70,61,0,19
Action: Stop Data Backup Process
Assessing threat severity...
[Weighted Threat: 125] [Default Severity: M] [Change Made: Increased by 1 Level] - [New Severity: H]
Outcome 1: USER REQUEST REJECTED
Outcome 2: USER ACCOUNT LOCKED ONLY USED SYSTEM UNTIL ALLOWED BY LEADERSHIP
Outcome 3: Attention Power User: Your account has been locked on this system. Action not allowed by
self-protected system without prior authorization.
Outcome 4: Alert sent to sup. & sec. team mobile device 3 times in 5 min
Outcome 5: Urgent email sent to sup. & sec. team: User account for employee Russell Morrison has been
locked on this system. Account can be unlocked if 3 ITMAM-assigned security members approve the unlocking
Outcome 6: Incident details have been stored in ITMAM for analytics & data mining purposes
```

Figure 4. Threat level elevated from Medium to High

Specifically, the user had an HR flag in his record and an unacceptable performance rating. In spite of being an experienced user, he intentionally attempted to disable system data backup, an action that was unapproved by the configuration change board. Data collected from external systems also revealed flags related to the misuse of Twitter by posting derogatory statements about the company. The aggregation and analysis of all the factors contributed to the elevation of the threat level. However, the algorithm did not elevate the threat to

Severe because the action initially had a Medium threat level if committed independently of other factors. Figure 5 depicts a situation where a Medium threat level was demoted to Low. The action was to ignore patching outdated software. The user had no HR flag in his record, and he had an excellent performance rating. However, company-related systems indicated that the user had submitted a grievance case against one of his peers. Since the action itself was approved by the configuration change board, and the initial threat level was not significantly exacerbated by other factors, demoting it to a lower threat level was justified.

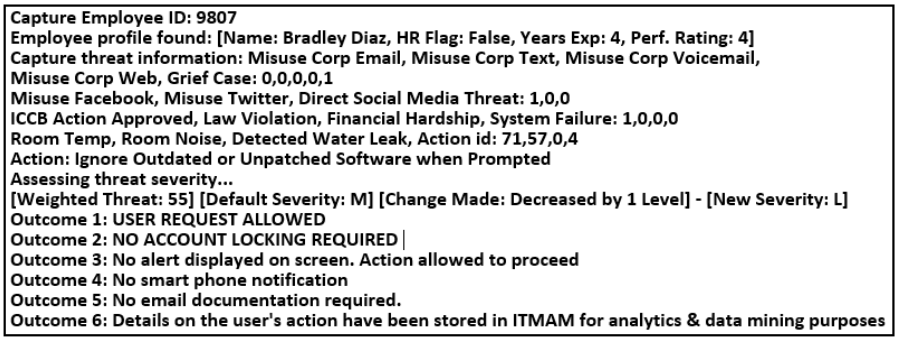

Figure 5. Threat level demoted from Medium to Low

Finally, Figure 6 illustrates how a threat level of Low was left unchanged. This inexperienced user had no HR flag in his record and had an excellent performance rating. Also, the action was approved by the change board, and a minor system failure was recorded. Although there was a misuse of corporate technology, the offense was not significant enough to warrant an elevation of the overall threat level.

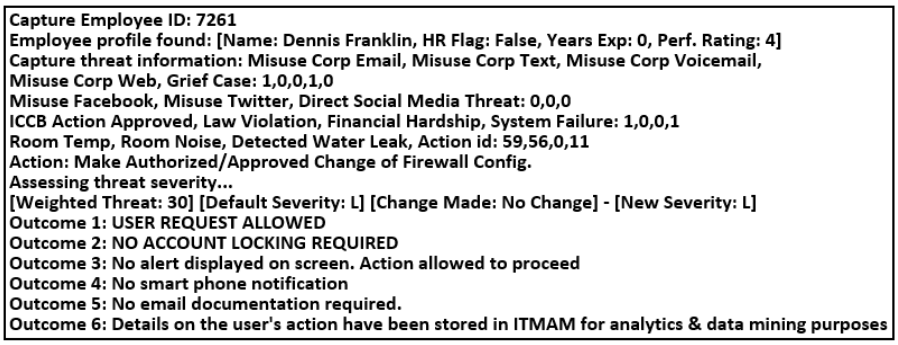

Figure 6. Threat level of Low kept unchanged

In addition to self-protection, the ITMAM framework incorporates business intelligence capabilities that deliver actionable information for making informed decisions. An analysis of the complete dataset showed a few relevant findings that a self-protected system will use to avert future actions as they progress. One finding reflected the correlation between the user's level of experience and the threat level. Assuming that the organization has a mostly experienced workforce, a Gaussian distribution was used to assign users a level of experience. The distribution was given a mean of 4 years and a standard deviation of 2. Scanning the full dataset revealed that only around 11% of the High or Severe threat level cases were attributed to inexperienced employees. This implies that the algorithm associated most of the insider attacks with more experienced users. Another finding linked the number of times a threat was elevated to a higher risk level with the decision rendered by the configuration change board regarding the action. In 35% of the recorded instances, the threat was elevated to a higher level at the time when the action was rejected by the configuration change board. This finding implied that power users, in our simulation, intentionally attempted to make a change to the system when they were explicitly told not to do it.

7. CONCLUSION

The paper introduced an autonomic framework designed to equip information systems with self-defense capabilities, to protect against a malicious action by an insider such as a system administrator. It simulated how an ITMAM-equipped system can protect itself against malicious actions by insiders. The paper also argued that such an approach is significantly more effective than other approaches that place the protection mechanism outside the system that is being protected. Simulation of a dataset of 150 records demonstrated how the system is able to protect itself against unauthorized actions. Future research related to this work will focus on building a more granular level of interdependencies between the weighted factors assigned to each action. Also, attention will be devoted to advancing the predictive analysis algorithm to establish a continuously growing intelligence base. This will allow the self-protected system to recognize early signs and extrapolate its self-defense attributes to avert an attack well before its occurrence.

REFERENCES

Ali, G., Shaikh, N. A., Shaikh, Z. A. (2008) Towards an automated multiagent system to monitor user activities against insider threat. *2008 International Symposium on Biometrics and Security Technologies, Islamabad*, 1–5.

BBC News. (2018) *Marriott hack hits 500 million Starwood guests* https://www.bbc.com/news/technology-46401890

Beena, A. L., Humayoon Kabir, S. (2019) Information Security Insider Threats in Organizations and Mitigation Techniques. *2019 International Conference on Recent Advances in Energy-Efficient Computing and Communication (ICRAECC)*

Buford, J. F., Lewis, L., Jakobson, G. (2008) Insider threat detection using situation-aware MAS. *2008 11th International Conference on Information Fusion*

Chen, Q., Lambright, J. (2016) Towards Realizing a Self-Protecting Healthcare Information System. *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*

Claycomb, W. R., Huth, C. L., Phillips, B., Flynn, L., McIntire, D. (n.d.). (2013) Identifying Indicators of Insider Threats: Insider IT Sabotage. *2013 47th International Carnahan Conference on Security Technology (ICCST)*

Ikany, J., Jazri, H. (2019) A Symptomatic Framework to Predict the Risk of Insider Threats. *International Conference on Advances in Big Data, Computing and Data Communication Systems (ICABCD)*

Jabbour, G., Menasce, D. (2009) Stopping the Insider Threat: The case for implementing integrated autonomic defense mechanisms in computing systems. *International Conference on Information Security and Privacy (ISP-09)*

Jabbour, G., Menasce, D. (2009) The Insider Threat Security Architecture: A framework for an integrated, inseparable, & uninterrupted self-protection mechanism. *2009 International Conference on Computational Science & Engineering*

Kim, A., Oh, J., Ryu, J., Lee, K. (2020) A Review of Insider Threat Detection Approaches with IoT Perspective. *IEEE Access*, *8*, 78847–78867

Mavroeidis, V., Vishi, K., Jøsang, A. (2018) A Framework for Data-Driven Physical Security and Insider Threat Detection. *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*

Paci, F., Fernandez-Gago, C., Moyano, F. (2013) Detecting Insider Threats: A Trust-Aware Framework. *2013 International Conference on Availability, Reliability and Security*

Schoenherr, J. R., Thomson, R. (2020) Insider Threat Detection: A Solution in Search of a Problem. *2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*

Schulze, H. (2019) *2019 INSIDER THREAT REPORT*. Cybersecurity Insiders

Schwartz, M. J. (2018) *Tesla Accuses Insider of Stealing Gigabytes of Data—Former Employee Accused of Hacking*

Software as Tesla Warns of "Sabotage." https://www.bankinfosecurity.com/tesla-lawsuit-alleges-insider-stole-gigabytes-data-a-11118, June 21, 2018

Verizon. (2019) *2019 Data Breach Investigations Report*. https://enterprise.verizon.com/resources/reports/2019-data-breach-investigations-report.pdf

Weise, E. (2018) *SunTrust said employee worked with outside criminal when info on 1.5M clients was breached*. USA TODAY. https://www.usatoday.com/story/tech/2018/04/20/many-1-5-million-accounts-may-have-been-compromised-suntrust-banks/535687002/

Zhang, H., Ma, J., Wang, Y., Pei, Q. (2009) An Active Defense Model and Framework of Insider Threats Detection and Sense. *2009 Fifth International Conference on Information Assurance and Security*