7-19-2017

A Technical Solution to Insider Threat Prevention

Vamsi Krishna Polam

A TECHNICAL SOLUTION TO INSIDER THREAT PREVENTION

by

Vamsi Krishna Polam

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

Major: Computer Science

The University of Memphis

August 2017

Acknowledgment

Firstly, I would like to thank my advisor, Dr. Dipankar Dasgupta, for his unconditional encouragement and support, for guiding me throughout the academic and research world, and providing me an opportunity to work with him and his team, trusting me in my work and advising me at every instance in the journey of my Master's career. I would also like to thank the members of my committee, Dr. Deepak Venugopal and Dr. Kan Yang for their rigorous support in providing specific comments and feedback which helped me in improving the content aspect of this thesis.

I am greatly thankful to Dr. Debasis Ghosh, especially for his help with the problem I am solving and patiently teaching me how to write with clarity. I am grateful to all my colleagues in our research group at the Centre for Information Assurance, University of Memphis. Our meetings and discussions helped me develop the skills to tackle certain problems. I am very grateful to all my professors and colleagues in Computer Science department for their constant support during my years at the University of Memphis.

Finally, I will take this opportunity to thank my father Mr. Narasayya Polam who has helped me in every step of my life fighting against all odds, my mother Mrs. Malleswari Polam, who has always been my greatest supporter. Without their unconditional encouragement, love and support, I would have never started this life-changing journey. My sincere thanks to you all.

Abstract

Malicious insiders are a serious security challenge to every organization due to their intimate knowledge of the organization's valuable information assets, resources and privileged access to those resources. Every organization needs to implement a defensive security policy to safeguard themselves from these security risks. To address insider threat problems, we implemented a framework which establishes trustworthiness among the employees based on Multi-user approval strategy. The framework uses a hierarchical structure of the employees in the organization such that if any user requires access to certain classified information, the framework selects a set of approvers randomly and sends the request to those approvers who are authorized to grant permissions. We implemented an application based on this framework to accomplish the goal and a thorough performance analysis is conducted to arrive at the result.

Table of Contents

List of Figures

Figure **Page**

Introduction

Most of the organizations can detect and mitigate risks associated with an outsider (non-employee) who tries to access and steal an organization's sensitive information. However, the attacker is difficult to detect and could cause huge amount of damage being the insider. We can define an insider as an individual who has or had legitimate and authorized access to an organization's information assets and resources and use the available access, either maliciously or unintentionally, in a way which negatively affects the organization (CERT, 2017). Insider threat events do not occur occasionally when compared to external attacks, but when witnessed usually pose a much higher severity of risk to the organizations. We implemented a solution based on Multi-user approver strategy. We developed a framework which alleviates permission approval for employees to access sensitive information and increase the level of trustworthiness. Employees at a lower level can request access to sensitive information and an employee at a higher level(approver) can grant or reject a specific request.

Problem Statement

Many factors increase an organizations' exposure to threats posed by insiders, because of which technical controls are limited. To overcome such threats, organizations must develop deeper understanding of trust, and work towards improving the trustworthiness of insiders (ISF, 2015). However, there is no proper solution to address these issues. In this research, we tried to implement a framework to develop a trust among employees to access sensitive information based on the Multi-user approver strategy.

Motivation

In recent years, we have witnessed a growing number of enterprises and government agencies suffering data breaches due to insiders. Recent surveys and statistics explain the amount of risk posed by insider threats, and increased insider risks. According to (Ponemon, 2017), 62% percent of business users report that they have access to company data that they probably should not see. Considering companies which experience data breach, insiders were responsible for 43% of data loss, among which one half of them was intentional, and the other accidental (Seals, 2015). 90% of security professionals trust employees with privileged access most of the time but only 41% trust insiders completely (BOMGAR, 2017). Figure 1 illustrates the statistics related to the insider security risks.

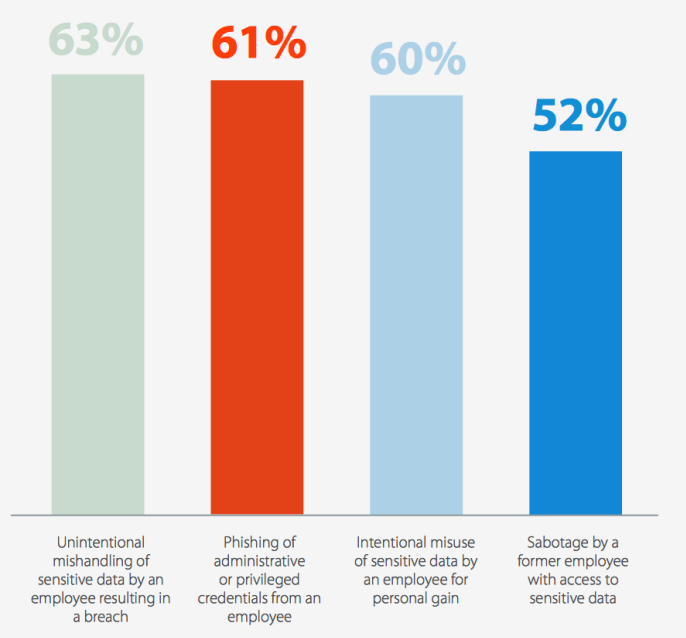

Figure 1. Insider Security Risk Statistics (BOMGAR, 2017)

We developed a solution to address trustworthiness among the employees through shared responsibilities. If any user requests for a sensitive file to be accessed, the framework will select the approvers randomly and send user's request to selected approvers. The approvers can accept or reject the request and notify the requested user. In this framework, all the activities among the requester and approvers are monitored through log files to analyze fraudulent behavior.

Limitation

In this study, we have chosen only two approvers per request. We implemented a hierarchical model to construct an organizational structure. Role-based access control methods can also be used instead of hierarchical model for creating rules of abnormal behavior for each rule. We have implemented limited classification of the files.

Roadmap

In the next section, we present a brief introduction to Insider threats and its effects on the organizations. Section "Multi-approver Strategy" illustrates several steps involved in the solution framework. Section "Implementation" provides an approach to the developed framework. The next section "Experiments and Results" will list out all the observations. Eventually, in the last two sections, we summarize the thesis highlighting future avenues on this line of research.

Background

We can define insider as an individual who has or had legitimate and authorized access to an organization's information assets and resources and use their access, either maliciously or unintentionally, in a way that could negatively affect the organization (CERT, 2017) . Insider threats pose a serious concern to various industries like government organizations, information technology, financial institutions etc. It is cumbersome to identify and monitor malicious insider's actions within the organization unlike the attackers from outside. Malicious insiders have advantage over attackers from outside as they have intimate knowledge of an organization's information assets and resources in addition to the authorized access to the system to execute malicious activities (Omar, 2015). Figure 2, adopted from the (CERT, 2017), depicts the type of individuals who can be considered as insiders and various assets that an insider can access along with the significant damages which can be caused to the organization.

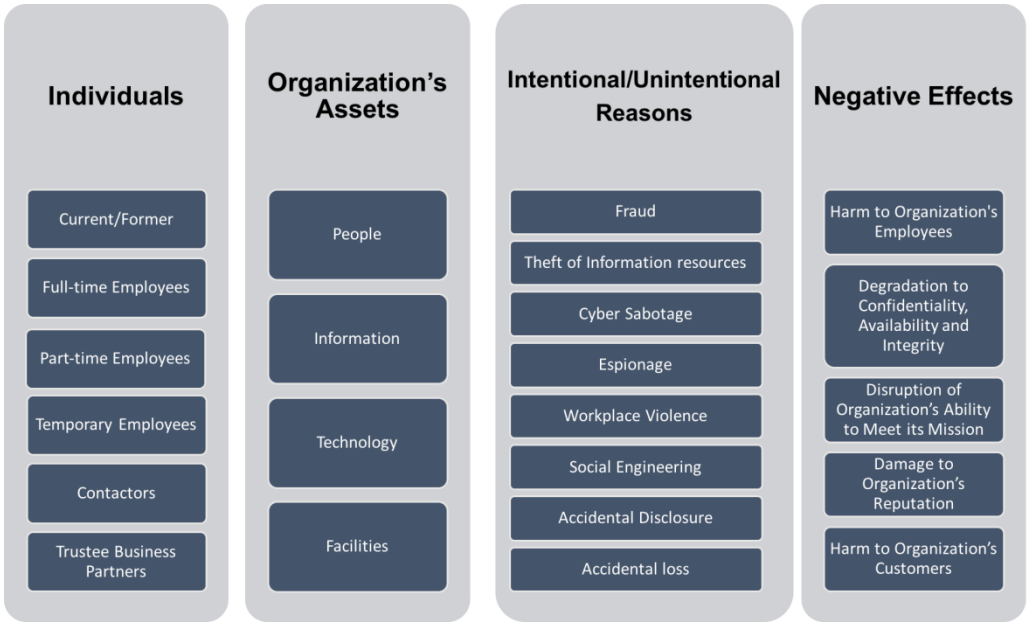

Figure 2. Factors Involved in Insider Threat Problem (CERT, 2017)

Insider attacks/Incidents

According to IBM, 55% of cyber-attacks were carried out by insiders. Security research also found that health care, IT, government and financial services are the top industries under attack, due to their personal data, intellectual property, physical inventory, and massive financial assets (IBM, 2016). When insider data breaches take years to discover then organization has lost control of sensitive data. Figure 3 depicts the breach discovery timeline of insider data breaches (Verizon, 2017).

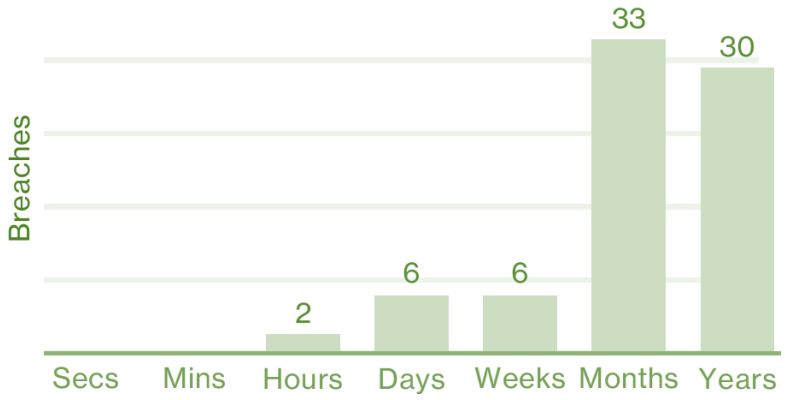

Figure 3. Breach discovery timeline within Insider and Privilege Misuse (Verizon, 2017)

Some of the incidents which involve insiders lead to a phase which indicates the severity of the insider landscape in every field of industry. Some of the incidents are mentioned below:

- Government agencies are also victims to insider threats. Some incidents in the past few years display a strong concern regarding insider threats. Edward Snowden, former technical assistant for the CIA, disclosed classified information in 2013 which contained NSA's domestic surveillance practices (Glenn Greenwald, 2013).

- Bradley Manning incident in which an employee from the US army disclosed sensitive information paved way to WikiLeaks. Manning was sent to Iraq as an intelligence analyst in 2009 through which he gained access to the sensitive military files for more than 700,000 documents, videos and diplomatic cables and sent to the anti-secrecy group, leading to WikiLeaks after which many of them were published. (ABC, 2017)
- Insiders who can gain access to confidential data post specific information on a website and attempt to sell the data in different markets. Morgan Stanley data breach is an example of such an incident. Morgan Stanley's employee Galen Marsh who worked as finical advisor from 2008, stole 350,000 records which had different client's sensitive data and posted them on Pastebin, a text-posts sharing website. (Schmerken, 2015)
- Jun Xie, A Chinese engineer who worked for GE Healthcare stole about 2.4 million files containing trade secrets and other confidential company information and sent it to China. He downloaded huge amounts of materials which were important to GE Healthcare and copied it to a separate storage device. He stole millions of files consisting trade secrets related to engineering designs, testing data, business strategy and source code for magnetic resonance systems from GE Healthcare (Vielmetti, 2014).

Cases of trusted insiders who abused their privileges to steal data include Manning Breach GE Health care incident, Morgan Stanley breach and Edward Snowden incident, which highlight the increasing need for better security practices and solutions to reduce the risks posed by insider threats. Organizations are inspecting their networks and

system for intruders, viruses, and malware, but the latter is another threat landscape leading to the loss of company assets.

Type of Insiders

Insider threats can arrive from different idealists like Snowden and Manning who expose an organizations sensitive information for personal benefits. The threats can also come from employees who are motivated by monetary benefit. Some insider threats are caused by non-malicious behavior which is a result of carelessness or lack of competence. We can classify the insider depending on their motives. According to (Flynn, 2012). there are mainly two types of insiders malicious and non-malicious insider as shown in Figure 4.

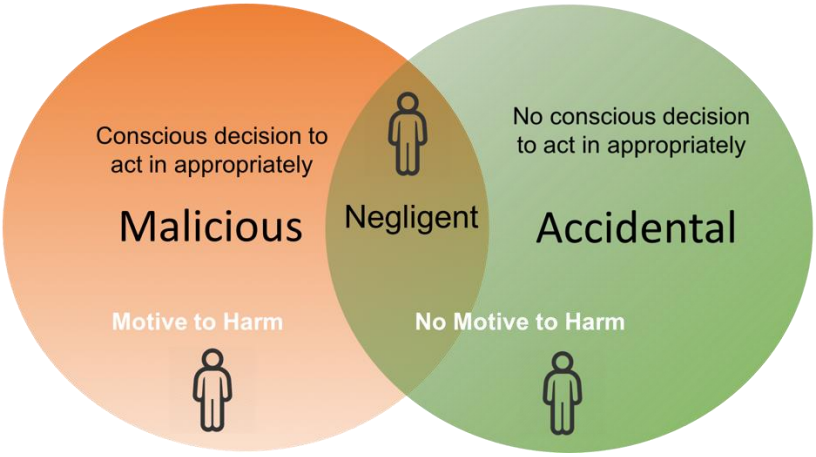

Figure 4. Types of Insiders

- **Malicious Insider -** Some employees try to expose an organizations sensitive information resources intentionally due to personal or monitory gains, revenge, or any other reasons. A disgruntled employee who resigned from the company but still has access to old privileges or with a motive to attack the company's data

creates back doors before leaving the company are all classified as malicious insiders. Sometimes, outside parties approach a trusted employee in the organization with an offer of monetary gain in exchange to insider sensitive information. As recently as 2013, a US soldier, Colton Millay, was sentenced for trying to sell secrets to Russia (Woolley & Troutman, 2014).

- **Negligent insider -** Another category of insider threat is a negligent insider who does not intend to harm. An employee who responds to phishing emails and disclose confidential information or lose laptops, mobile devices and pen drives or other devices that contain confidential information are classified under this category. An inattentive, careless, poorly trained employee can expose sensitive information and fall prey to the adversary traps. IRS employee accidentally expose thousands of government employees to identity theft. Cyber criminals use social engineering attacks like spear phishing on employees to gain unauthorized access to the organization's resources (Woolley & Troutman, 2014).

Traditional Security policies

Organizations mostly depend on security policies, auditing log monitoring tools, and traditional access control mechanisms to address insider threat issues. But these techniques are unable to resist emerging insider threats which are highly sophisticated and those which usually do not leave a black spot after an insider attack. A root cause of the insider threat issue arises from business organizations, and government agencies due to inadequate security defenses in place to detect and prevent insider attacks (Omar, 2015).

Most of the organizations invest money in securing information and infrastructure against outside malicious attacks and shower less focus on the threats posed by insiders who can advertently abuse privilege access to an organizations information assets and steal sensitive business data for malicious purposes (Barrios, 2013).

For example, most organizations depend on log files to monitor employee activities but it is tedious to interpret accurate malicious behavior. Although current systems maintain user access logs and activity logs, it can easily be deleted by malicious users. Highly skilled rouge super-users can erase these log files to remove the trace of their malicious activities, making off-line analysis ineffective.

Multi-User Approver Strategy

Multi-user approver strategy is a solution to prevent insider data breaches through shared responsibilities such as user activities with classified data in an organization where sensitive data is being accessed. They are to be regulated via an approval process so that others are aware of such an access. Such a permission strategy should consider employee's role and hierarchy while determining the approvers. Any request for accessing a classified document needs approval from a set of users. After receiving a request from a user to classified files, it determines the number of approvals as well as find the appropriate approvers and notify them accordingly (Dasgupta, 2015).

Multi-User Approval Framework

This framework allows employees of an organization who require access to files of higher sensitivity to request permissions from employees who are at a higher level in comparison with them. Employees at the higher level can grant permission to specific

employees who request an access as they are promoted to be the approver for that specific file. Multi-User Approval strategy consists of several steps to arrive at a shared trust between the requester and approver to access any classified information (Dasgupta, 2015). The steps involved in this framework are as described below.

Step-1: Model Organizational Structure and Classification of files

Files are classified and archived based on their level of sensitivity. The framework selects the number of permissions needed to access the files based on the sensitivity of those files. Framework uses the government classification for the files such as top secret, secret, confidential, restricted, and unclassified for applying a multi-user permission strategy. The United States government classifies information according to the degree which the unauthorized disclosure would damage national security. A classification level as shown in Figure 5, indicates the importance of classified information to national security and thereby determines the specific security requirements applicable to that information (FAS, 2017).

Figure 5. Levels of File Classification Used by U.S.A Government

Employees in an organization are classified into different hierarchical levels based on their designation/roles. These employees are provided with suitable permissions to

approve files based on their level. Figure 6 describes the organizational structure and classification of files.

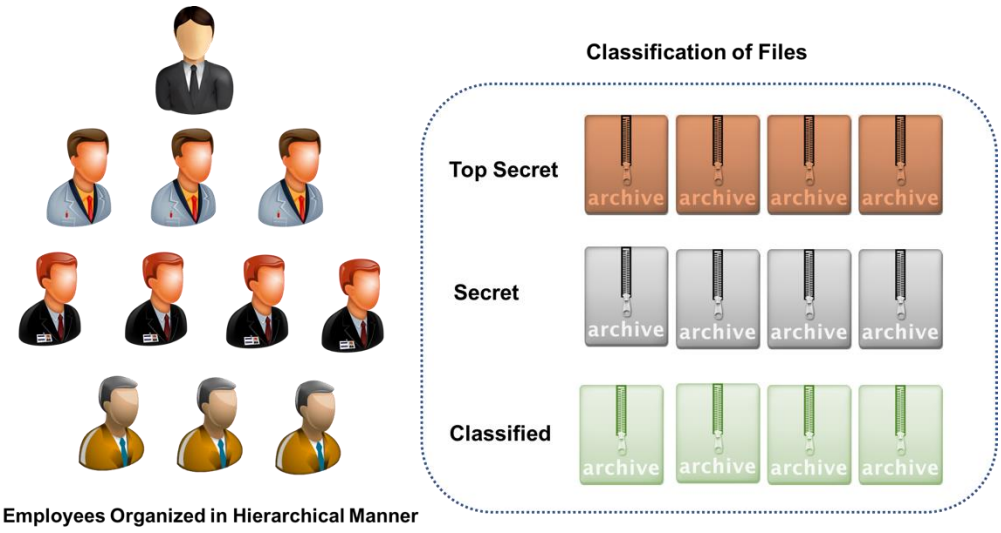

Figure 6. Organizational Structure and File Classification

Step-2: Employees at a lower level can request an employee at a higher level for permission to access a sensitive file.

Employees at a lower level cannot access files with higher sensitivity. If an any employee requires access any file which has a high sensitivity then they should request the employees at higher level for permissions. Figure 7 illustrates the request process to access classified files by the users.

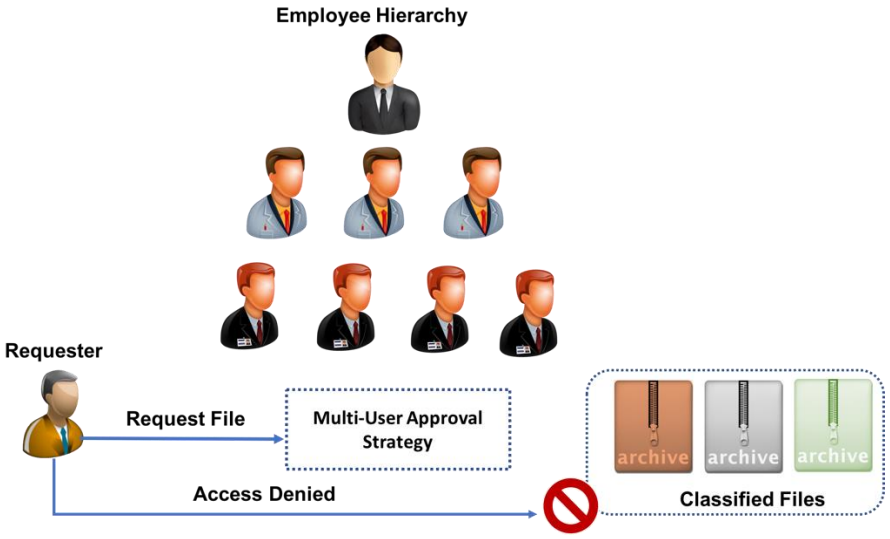

Figure 7. Requesting for the Classified Files by the Requester

Step-3: Randomized approver selection

The request from the employees is processed in this step. Based on the request from the employees the framework selects some approvers randomly, selecting from a set of different approvers at the higher level. Figure 8 describes the randomized approver selection process.

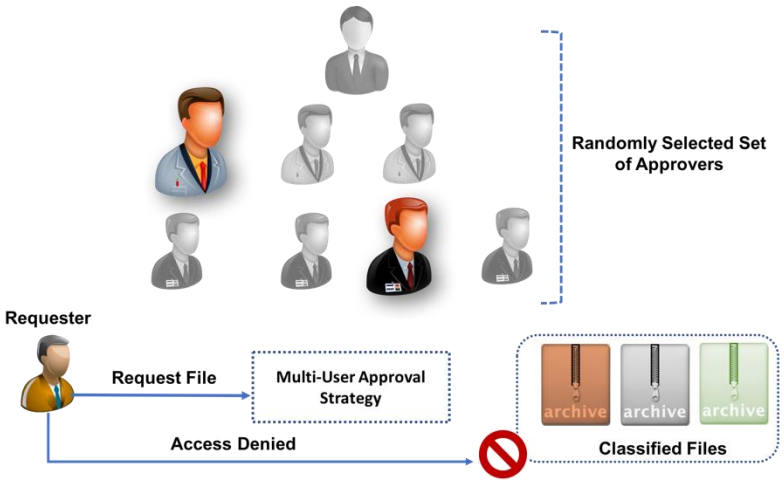

Figure 8. Randomized Approver Selection

Step-4: Selected approvers are sent request approvals

After selecting the approvers, the framework will send a request notification to those employees who are selected as approvers in the previous step. Figure 9 illustrates notifying approvers regarding the request.

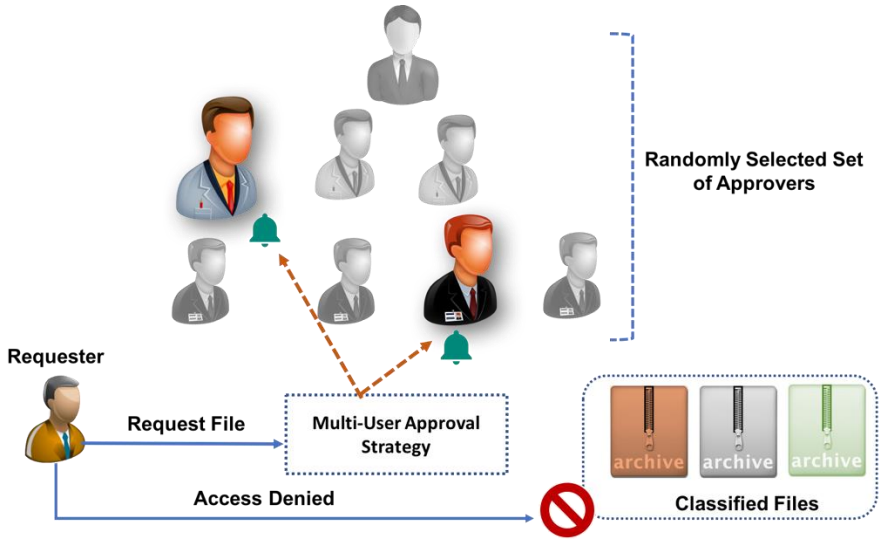

Figure 9. Sending Notifications to Selected Approvers

Step-5: Acceptance/Rejection by the approver

After getting the request notification from the requester, the approver can accept or reject. If it is accepted an approval notification is sent to the requester. If it is rejected a notification along with the reason for rejection is sent to the requester. Figure 10 describes the approvers decision process regarding the received request.

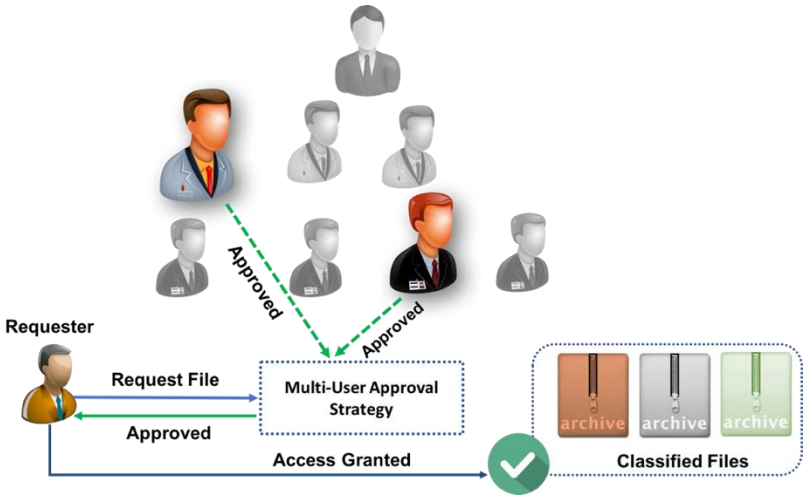

Figure 10. Approval Process

Step-6: Create logs at Requester and Approver side

The framework automatically creates two log files which are requester logs and the approver logs to monitor and generate reports for the activities involved between the request and the approve process as shown in Figure 11.

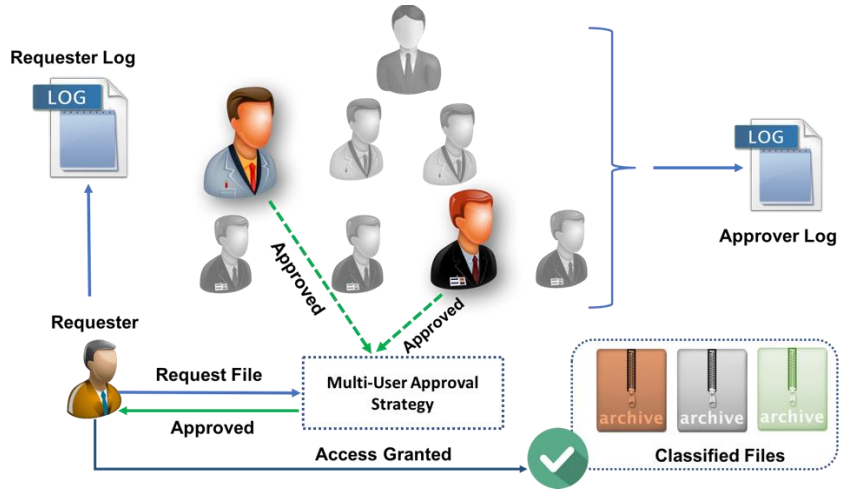

Figure 11. Activities Monitoring Through Logs

Implementation

We implemented an application based on the multi-user approval framework to display the activities between the requester and the approver. We created different user accounts for the employees to request and approve more interactively. We used various tools and commands in the Linux environment to accomplish the tasks.

Environment setup

We developed the application on a Linux system with 8GB memory and 256GB secondary memory. The following list describes the technical requirements to develop this application:

- Environment: Linux Operating System (Ubuntu)

- Backend: MySQL database
- Programming Language: Bash Scripts
- Front-End: YAD and Zenity scripts

We created an employee table on MySQL server which the application can access and select the appropriate approvers for the request. Figure 12 depicts the ER diagram for the employee table.

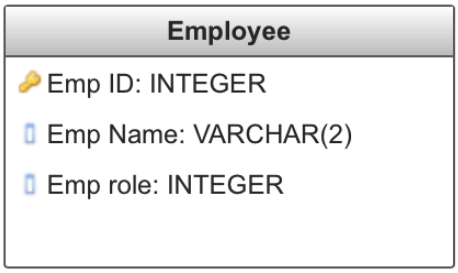

Figure 12. ER Diagram for Employee table

Application

We developed an interactive front end for the requester and approver through which users can easily interact with the application.

Requester Side

This application consists of four buttons with different tasks at the requester side. Figure 13 depicts the front end for the requesters.

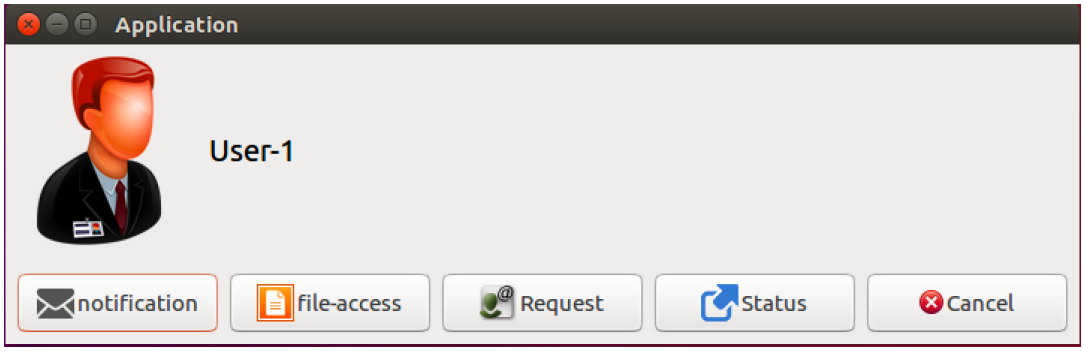

Figure 13. Frond end for requester to interact application

Notification: Requester can visualize the notifications from approvers

File- access: Requester can view the classified files and check for accessibility

Request: Requester can send file access request

Status: Requester can check the status of the request

Figure 14 depicts the features provided to the requester to accomplish all the tasks mentioned above.

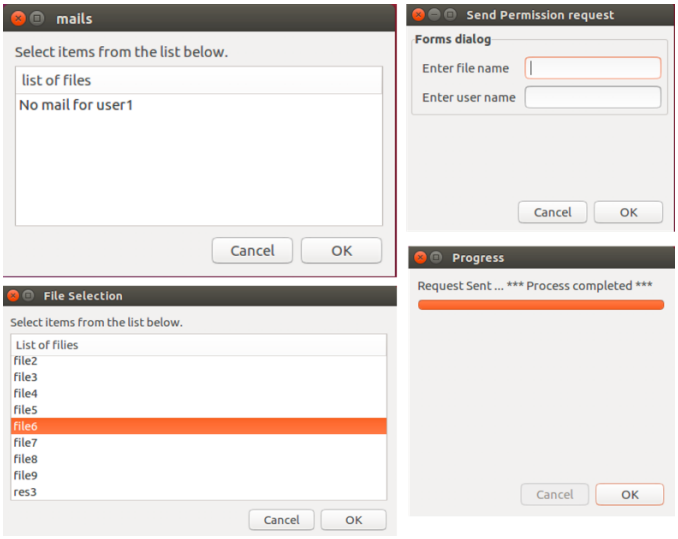

Figure 14. Features in Requester window

The requester can select the files and send a request permission to the approvers as mentioned in above figure. The application will analyze the request made by requester based on the hierarchal level of the employee and sensitivity of the file and select two approvers randomly according to the framework logic as mentioned in the previous section. Figure 15 illustrates the approvers selection in the application.

Figure 15. Approver's selection

Approver Side

This application consists of four buttons with different tasks at the approver side.

Figure 16 depicts the front end for approvers:

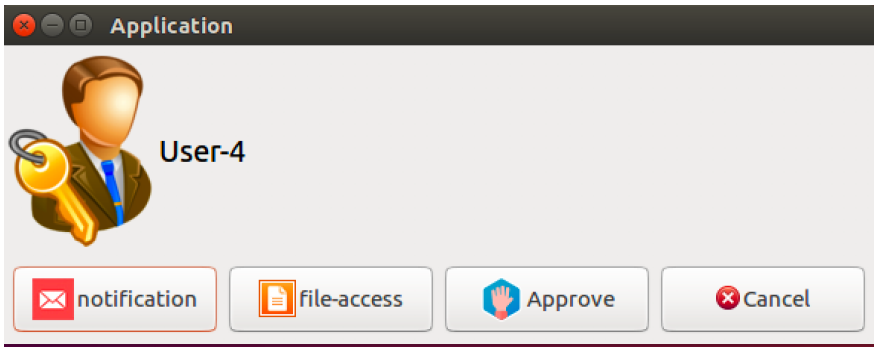

Figure 16. Window for Approvers to Interact Application

Notification: Approvers can access the notifications from requesters

File-access: Approvers can access the files

Approve: Approvers can approve/ reject the requests made by the requester.

Figure 17 depicts the features in the approvers side application to accomplish the tasks.

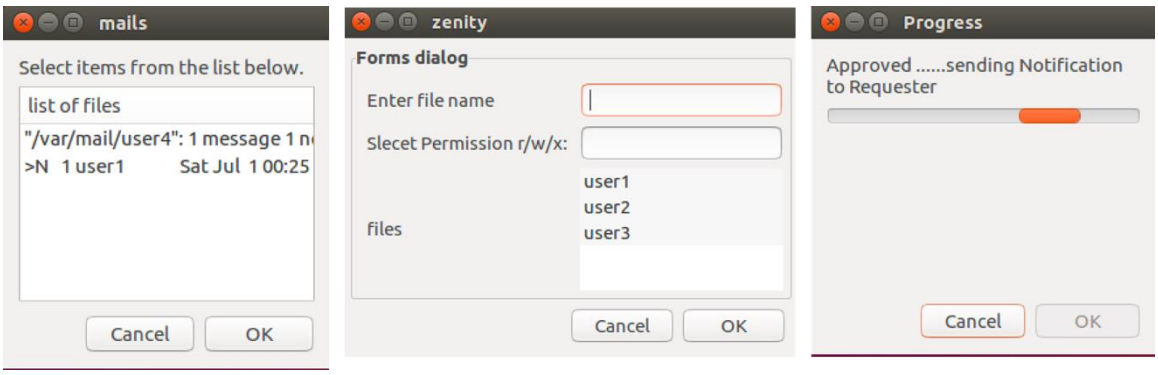

Figure 17. Features in Approvers Window

The application monitors all the activities including the request-approve process regularly through log files. It creates two log files named requester log and approver log to monitor both requester and approver activities. Figure 18 depicts the logs generated by the application.

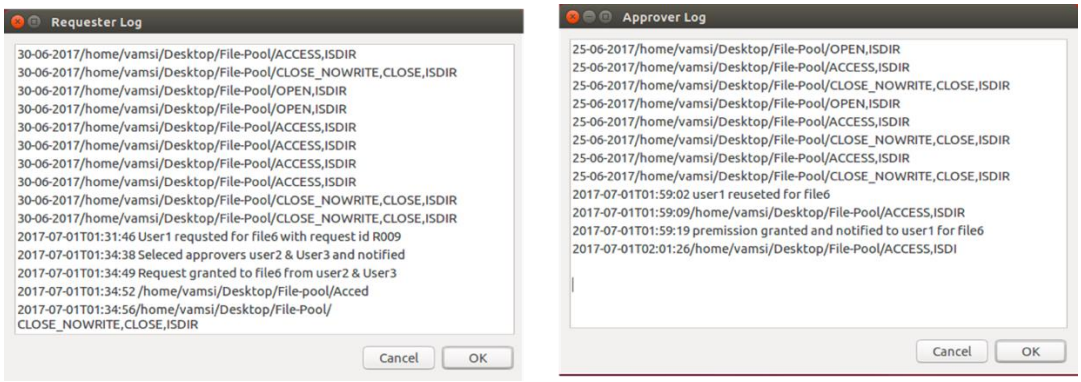

Figure 18. Generation of Log files

Empirical Results and Evaluation

We conducted experiments to evaluate the performance of the application. This performance technique will be helpful in deciding if the implementation is successful. Sample data for 100 employees is considered and it is stored in the MySQL database. We created 100 user accounts and a sample file system which consists of 20 files and they are classified into two levels namely top secret and secret.

We conducted different experiments with different requests from the employees table in the database and corresponding user accounts. We recorded the time taken to select approvers along with sending notification to approvers and calculated average time taken. Also, we recorded the time take to approve a request by the approver. We conducted 20 experiments as shown in Table *1*, for requests R1 to R20 (20 requests).

Table 1.Time Measures from the Experiments

Request ID	Requester	File ID	Approver-1	Approrver-2	permission status	Time taken to Request (Sec)	Time taken to Approve (Sec)	Total Time taken to complete Request (Sec)
R1	E9	F1	E21	E32	Yes	0.168	0.122	0.29
R2	E3	F7	E6	E44	Yes	0.152	0.116	0.268
R3	E15	F4	E17	E66	Yes	0.166	0.121	0.287
R4	E5	F6	E56	E81	Yes	0.156	0.188	0.344
R5	E3	F9	E33	E5	Yes	0.181	0.133	0.314
R6	E10	F7	E18	E44	Yes	0.161	0.155	0.316
R7	E1	F1	E7	E62	Yes	0.151	0.112	0.263
R8	E22	F3	E46	E62	Yes	0.184	0.12	0.304
R9	E45	F4	E51	E89	Yes	0.155	0.122	0.277
R10	E5	F6	E7	E66	Yes	0.162	0.199	0.361
R11	E16	F3	E44	E31	Yes	0.145	0.129	0.274
R12	E89	F6	E91	E95	Yes	0.132	0.149	0.281
R13	E36	F5	E39	E45	Yes	0.155	0.116	0.271
R14	E78	F2	E81	E96	Yes	0.161	0.104	0.265
R15	E35	F9	E53	E75	Yes	0.152	0.221	0.373
R16	E43	F4	E63	E54	Yes	0.162	0.115	0.277
R17	E39	F1	E47	E81	Yes	0.166	0.116	0.282
R18	E31	F9	E53	E72	Yes	0.153	0.144	0.297
R19	E42	F6	E64	E45	Yes	0.155	0.155	0.31
R20	E66	F7	E71	E84	Yes	0.148	0.128	0.276

Table *1* depicts the different time measure based on the results of our experiments. The columns indicate the requester ID, specific file requested by the requester (Requester file), the ID of the file, specifics of the approvers, the permission status (Yes or No) indicating if the request was approved or denied, amount of time taken to request a file, amount of time taken to approve a file, total time taken to complete the request.

Table 2. Time measures for Request Completion

Average Time Taken to Request	0.15825 Sec
Average Time Taken to Approve	0.13825 Sec
Average Total Time Taken to Complete Request Process	0.2965 Sec
Standard Deviation on Total Time	0.0316 Sec

We plotted two graphs related to the average time taken to request and average time taken to approve for the 20 experiments (R1- R20) as shown in Figure 19 and Figure 20 below.

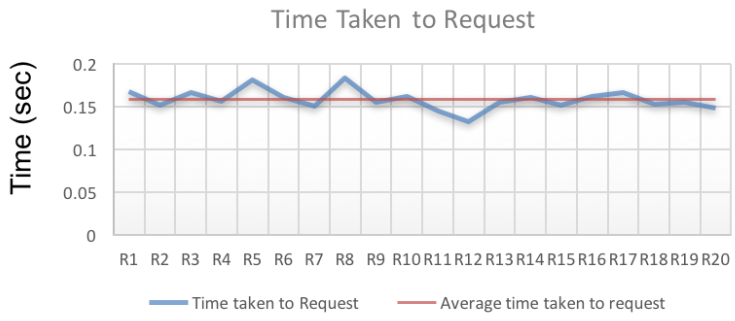

Figure 19. Time Taken to Request

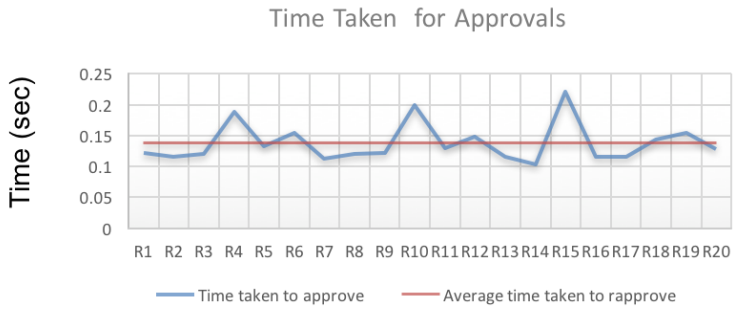

Figure 20. Time taken to approve

We calculated the meantime to complete one request (sending request and approved by approver). Also, measured the standard deviation for all requests to analyze data behavior as shown in Table *2*. If the data behaves in a normal curve, then 68% of the data points will fall within one standard deviation of mean data point. More variances cause more data points to fall outside the standard deviation. Smaller variances result in more data that is close to average. In our experiments, we are assumed the approvers will grant the approval immediately after getting the request from the requester.

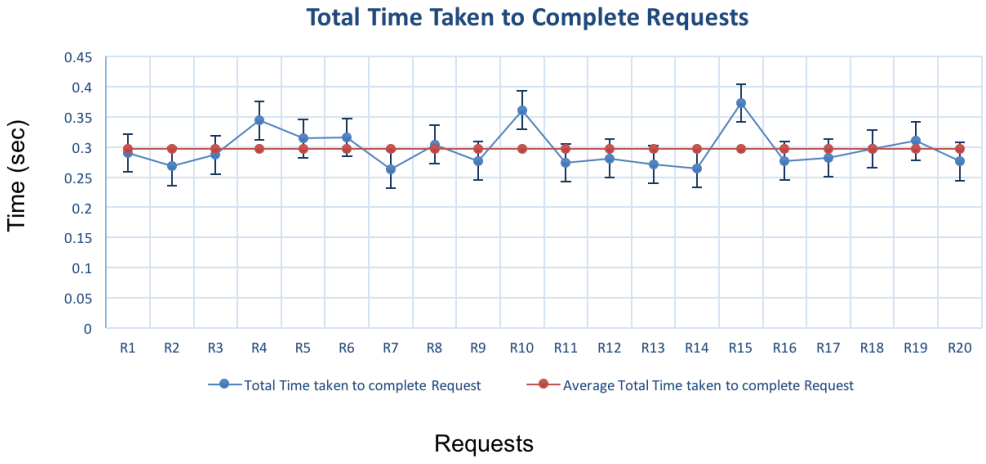

Figure 21. Performance Analysis Graph

From the above graph depicted in Figure 21, we can analyze the data behavior. Like most data, requests with typical time consumption probably turn out to be normally distributed. That is, for most requests the time consumption will be close to the mean, while fewer requests take more time than the mean. We considered one standard deviation to measure the variations.

This analysis helps in fully analyzing the fraudulent patterns which may occur in the process of requesting and approval. The requests which have taken more time to complete the process at regular intervals can be treated as a variant behavior which results in the attacker consuming time to act maliciously and steal some information, modify or delete the malicious footprints permanently from the system.

Conclusion

Insider threat problem pose a great risk to every organization but organizations lack an implementation of strong security framework to defend against them. We implemented an application based on Multi-user approval strategy to address insider threat problems. We considered hierarchical model of organizational structure and classified files based on their level of sensitivity to arrive at the framework. The framework allows a user to request classified files and the request will be sent to randomly selected approvers who are provided with the authority to approve. To monitor all the activities in this permission and approver model a set of log files are automatically generated at regular intervals both on the requester and the approver side. We conducted performance analysis on this application based on the total time taken to complete a request.

References

ABC. (2017, Jan). *Chelsea Manning: Who is the convicted military leaker whose sentence has been slashed by Barack Obama?* Retrieved June 15, 2017, from ABC News: http://www.abc.net.au/news/2017-01-18/who-is-chelsea-manning/8190214

Barrios, R. M. (2013). A Multi-Leveled Approach to Intrusion Detection and the Insider Threat. *Journal of Information Security*, 54-65.

BOMGAR. (2017). *THE SECURE ACCESS THREAT REPORT 2017.* BOMGAR. Retrieved June 15, 2017, from BOMGAR: https://dbac8a2e962120c65098-4d6abce208e5e17c2085b466b98c2083.ssl.cf1.rackcdn.com/mitigating-risks-from-privileged-insiders-vendors-pdf-2-w-3436.pdf

CERT. (2017). *Insider Threat Blog*. Retrieved June 10, 2017, from CERT Carnegie Mellon: https://insights.sei.cmu.edu/insider-threat/2017/03/cert-definition-of-insider-threat---updated.html

Christopher Woolley, M. D. (2014). *Insider Threat: Policy Impact and Overview.* Center for Infrastructure Protection and Homeland Security George Mason University School of Law. Cyber Security and Information Systems Information Analysis Center.

Dasgupta, D. (2015). Multi-user Permission Strategy to Access Sensitive Information. *or Peer Review Multi-user Permission Strategy to Access Sensitive Information Journal: Transactions on Information and System Security*.

FAS. (2017). *Security Classification of Information*. Retrieved June 10, 2017, from Federation of American scientists: https://fas.org/sgp/library/quist2/chap_7.html

Flynn, G. S. (2012). *Common Sense Guide to Mitigating Insider Threats 4th Edition.* CERT Program.

Glenn Greenwald, E. M. (2013). *Edward Snowden: the whistleblower behind the NSA surveillance revelations*. Retrieved June 5, 2017, from the guardian: https://www.theguardian.com/world/2013/jun/09/edward-snowden-nsa-whistleblower-surveillance

IBM. (2016). *Reviewing a year of serious data breaches, major attacks and new vulnerabilities.* IBM X-Force. Retrieved from Reviewing a year of serious data breaches, major attacks and new vulnerabilities: http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=WH&infotype=SA&htmlfid=SEW03133USEN&attachment=SEW03133USEN.PDF

ISF. (2015). *Managing the Insider Threat: Improving trustworthiness*. (S. Durbin, Producer) Retrieved June 15, 2017, from Informa on Security Forum.

Omar, M. (2015). *Insider Threats: Detecting and Controlling Malicious Insiders.* IGI Global.

Ponemon. (2017). *Data Theft Rising Sharply, Insider Threats Cited as Leading Cause*. Retrieved June 21, 2017, from Ponemon Institute: https://www.varonis.com/learn/ponemon-2016/

Schmerken, I. (2015). *Morgan Stanley Data Theft Exposes Insider Threat & Need for More Restrictions*. Retrieved June 10, 2017, from Wall street and Technology: http://www.wallstreetandtech.com/security/morgan-stanley-data-theft-exposes-insider-threat-and-need-for-more-restrictions/d/d-id/1318623

Seals, T. (2015). *Insider Threats Responsible for 43% of Data Breaches*. Retrieved June 20, 2017, from InfoSecurity: https://www.infosecurity-magazine.com/news/insider-threats-reponsible-for-43/

Verizon. (2017). *2017 Data Breach Investigations Report 10th Edition.* Verizon .

Vielmetti, B. (2014). *Chinese engineer accused of stealing trade secrets from GE unit*. Retrieved June 20, 2017, from Journal Sentinel: http://archive.jsonline.com/news/crime/chinese-engineer-accused-of-stealing-trade-secrets-from-ge-unit-b99344912z1-274122821.html/