PRAGMATIC REFORM TO AMELIORATE INSIDER DATA THEFT DETECTION

K. Sri Sumanth[1],Sridevi Sakhamuri[2], K. Yaswanth[3], K. Revanth[4]

Koneru Lakshmaiah Education Foundation,

(Deemed to be University)

Guntur, Andhra Pradesh, India.

(sumanth.kommenni@gmail.com[1],sridevisakhamuri@kluniversity.in[2],kumbhayashwanth@gmail.com[3],revanthkowtharapu1999@gmail.com[4])

Abstract: Cyber threats are escalating moderately, and damage caused by them are priceless. We must face and respond to them in a gingerly manner. Insider attacks are very boorish, and their attack vectors may start from any corner in a company or an organization. The malicious insiders having authorization to access the sensitive assets makes the organization vulnerable and even make it permeable to hackers. They might cause physical damage, data theft, exposing business secrets and also lead to any other operations which are against the privacy policy of company or an organization. They must be picked out pragmatically, threats must be espied and necessary legal actions must be taken. Here we elucidate starting from the nuts and bolts of insider threats and theft to their detection and response. Our study will make it easy by providing the empirical evidence and the latest trends, techniques. Our study proposes a hybrid solution model named Signature based data theft detection and file monitoring system which is designed and proposed with utmost case scenarios concerned to defend and overcome against those threats.

Keywords: Attack, cyber threat, defense solution, insider theft detection, insider threat, signature-based theft detection, file monitoring system.

1 Introduction

Insider data thefts are increasing significantly, and they are hazardous to any organization. The IT and security teams like the blue team are hunting for the reasons behind those attacks. One of the main causes of the attacks is reckless and irresponsible employees. They either raise the attack vector or they might be behind the attack by supporting the hackers. The main problem is that many employees in an organization have proper and legitimate access to the assets and data. This makes Insider theft detection and prevention more challenging than external threats. Plenty of people is doing enormous research about Insider threats and thefts.
The following are the contributions made by this paper:

- To cope with insider threats and thefts, we propose a different way where we use a directory monitory system to monitor the directory continuously and alert the admin if there are any changes in it.
- A hash comparison technique that we use to compare the hash values of the files to detect insider data thefts.
- Our proposed approach has a considerably higher ability to detect insider data thefts because the analysis is based on directory monitoring and hash comparison.

1.1 Insider Threat Definition

An insider threat can be defined as evil threat that is caused by an individual who is an employee or a former employee or then who has a legitimate access to the assets of an organization, and they result bad outcomes for a company, an institution or an organization.

1.2 Insider Data Theft Definition

An insider data theft can be defined as stealing of data, assets or any sort of valuable information of a company or an organization that is stored in a server, databased or in other storage devices. It may be done directly by an insider or a compromised employee is used as an intermediator to perform an attack.

1.3 Insider Threat VS Insider Data Theft

Insider threats are those caused by an employee which makes the organization vulnerable. Among them, one of the possible menaces is data theft. It causes huge loss of money, trust, reputation, compromise of customer's data, disclosure of trade secrets, and face legal liabilities. The compromised employee's accounts lead the cybercriminals to lurk inside a network and stay unnoticed for long periods. The damage depends on the access rights of the compromised account. The motives of stealing data might be different from an individual, it might be to take vengeance from an employee or just gain some financial profit. There are different scenarios from which an insider can steal the data drawn from the report [1]. How insiders can steal data are demonstrated in Fig.1.

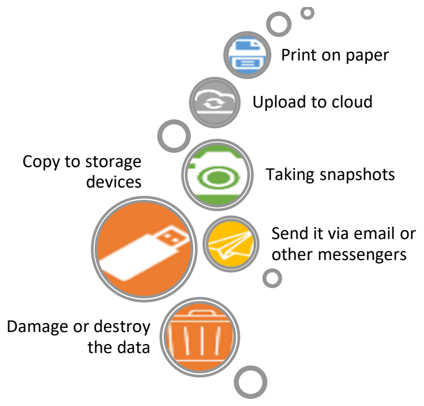


Fig.1. Ways in which an insider can steal the data.

1.4 Insider's chronicle

From recent studies, insider threats are costlier and more dangerous than external threats. In this section we will summarize the insider incidents, consequences adduced from surveys and technical reports. The reports [2],[3] states that 68 % of the organizations are moderate to extremely vulnerable to insider threats and confirm that insider attacks are more frequent than in earlier days. A majority of organizations (58%) consider themselves as least effective or even worse in monitoring, detecting, and responding to insider threats. They also released the types of insiders that pose the biggest risk to an organization categorized in Fig.2.

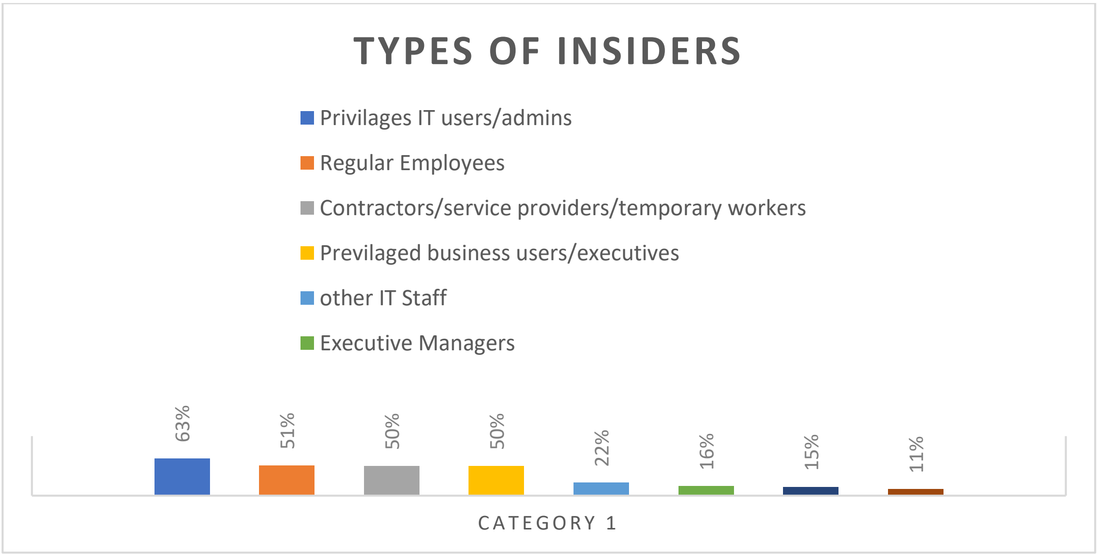


Fig.2. Types of insiders pose the biggest risk to organization.

Insider attacks area unit notably dangerous for 3 reasons:

• Insiders are not malicious all the time, but their dangerous activities lead to outside attacks.

• They create weakness in a company or an organization and make their IT security standards vulnerable.

• Insiders expose sensitive information which leads to the downfall of business reputation.

For these reasons, business executive attacks end in devastating losses for organizations. The outcomes of an insider attack are disclosed in Fig.3.

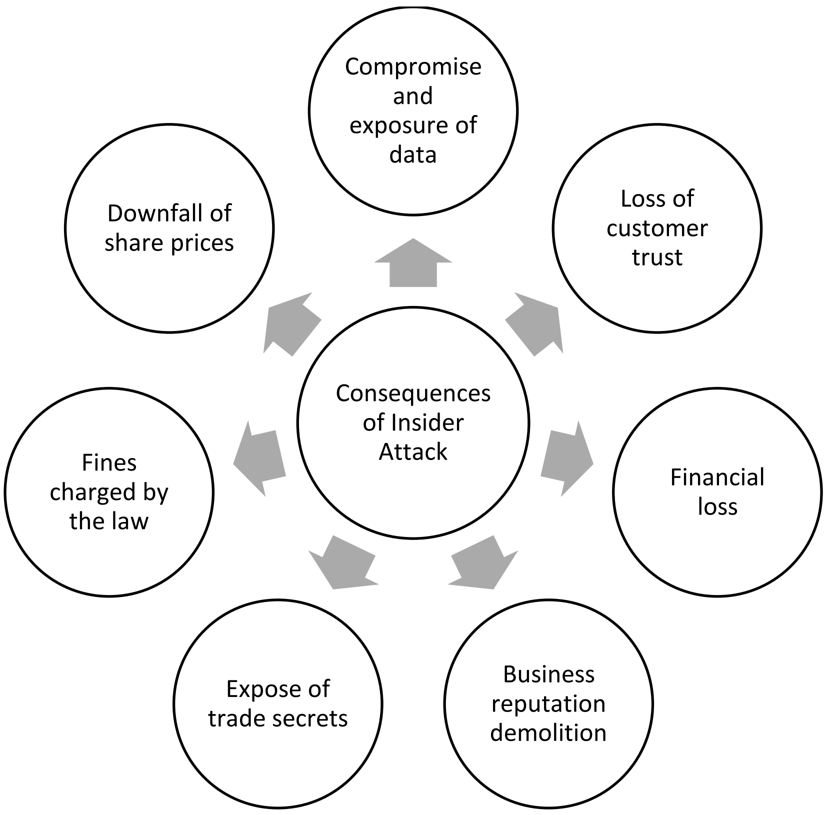


Fig.3. Possible Consequences of an insider attack**.**

Paper is categorized as follows: The section 2 narrates previous works, problem statement and proposed model is stated in section 3, results are in 4th section and the conclusion is concluded in section 5.

2 Related Work

Insider data thefts are increasing gradually, a lot of researchers are doing research and they are helping in mitigation those threats.
In [4], R. A. Alsowail and T. Al-Shehari have explained the types of insider threats, their detection techniques, and mitigating techniques. They had also explained the methodology to research insider threats to get efficient and better outcomes. The authors had also proposed a 10 - question model for a comprehensive study on insider threats with empirical factors. They had discussed the detection mechanisms like anomaly detection, Misuse based detection, and both combined detection. They also worked on CERT[12], RUU, SEA, ENRON datasets that are used for Insider threat approaches. Machine learning algorithms like SVMs, KNN, GMM that are employed in Insider threat detection are also hashed.

In [5], a UVM approach method is proposed to prevent data theft from cloud storage where the authors have developed a data theft prevention system and detection system which will work on the factors of what the user is searching and what file is opened. The verification system will work by using two mechanisms they are OTP verification and Question verification. This mechanism is for cloud systems and analyzing their results it shows that will enhance security.

In [6], an insider attack detection system is integrated with the packet filtering technique. Here the packets in the intranet are scanned and filtered, if any suspicious packets are found then they are dropped and the admin is alerted. In [7], a Role and User-Based profile assessed insider threat detection is designed in an automated way. Their framework comprises various key parts in which they process the approached log data record and then a client profile is developed along with the current day job. By this, the level of threat is evaluated for a person. In [8], a mechanism is proposed to mitigate insider attacks in the cloud environment. They implemented an alternate methodology to secure the information in the cloud by employing the innovation of hostile imitation. They planned it to guard data acquired in the cloud, recognize anomalous data access designs. If any suspicious unapproved access is detected, then it is verified by challenge question. If it's positive, then a lot of imitation data is sent to the masquerader.

The authors of [9] propose a new method for detecting malicious insiders that involves examining word similarity across multiple types of security records. The log2text component translates events from every type of security log into an identical format based on a 4W sentence template, and the text2corpus component organizes the changed events into a Word2vec trainable corpus by users and dates. From the data of [10], we can see that they have used an ML based system for insider threat detection in their intranet systems. They used LR, NN, RF, and XG ML algorithms on multiple levels of data granularity, limited ground truth and trained them in different scenarios to support the cybersecurity analysts to detect the unseen data of malicious insiders.

Although, the principal challenge in the insider threat region is data theft and assembling exceptionally exact frameworks that can anticipate, identify and forestall insider assaults consistently. The zone of insider threats and their protection arrangements is yet not seen quite well. A few overviews attempted to encourage the field by zeroing in solely on hypothetical and reasonable scientific classifications. Our aim is to minimize insider data theft by developing a mechanism to detect and monitor the activity continuously. In the following section, we have proposed a mechanism to detect and monitor insider data theft.

3 Methodology

In this section we illustrate about Descriptive Research Methodology [11] and it is encapsulated in Fig.4.

There are five stages in our methodology, and they are as follows:

Stage - 1:

Literature survey and definition: In the first stage, all the theory-related work about "Insider threats and thefts" is done to review the existing research. With the help of many search engines we query all the related information by using all the possible search terms (E.g., "Insider data theft detection", "Insider robbery detection", "Insider data breach detection" and "Insider threat detection technologies"). Furthermore, online blogs and related scripts are reviewed for utmost knowledge.

Stage - 2:

Publications and surveys: In the second stage, the related publications, research articles, and existing surveys are reviewed. Pedagogical sources like the Scopus database and IEEE XPLORE are used. Suitable search terms are queried that is specified. During our search, all the appropriate synonyms (e.g., cyberattacks, cyberthreats, vectors, issues, data theft, turncloaks, theft detection, etc.) are also reviewed in our study. Remember that not all our retrieved articles are not relevant. So, all the papers must be scrutinized by reading the title, abstract, conclusion, and some articles in them. We must include only those articles that have experimental results and empirical evidence. By this step, we will finish our theoretical approaches.

Stage - 3:

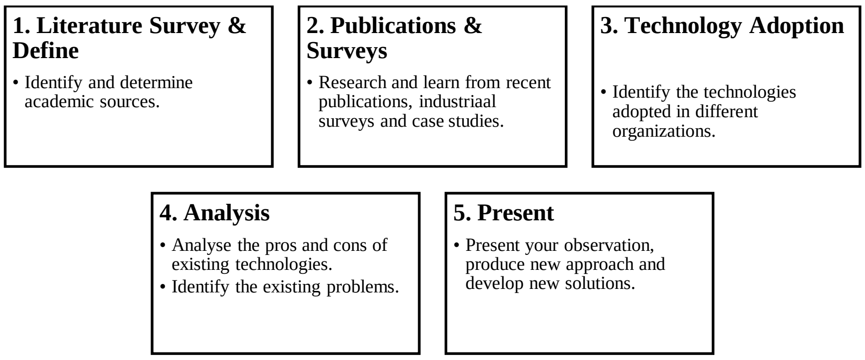
Technology Adoption: In the third stage, we identify the existing technologies that are adopted in an organization. Not only for one organization, but we also need to collect a plethora of data with empirical proofs and performance data. Identification of both the pros and cons of the technology is the crucial part. With the help of case studies, we can evaluate the better performing technique used in a company to detect a malicious insider. Almost all the available threat detection techniques are reviewed.

Stage - 4:

Analysis: In this stage, we analyze all the available technologies with their strengths and weaknesses, and ranked on the basis of their availability. In this study, we will identify the reasons and possible solutions to overcome the existing technologies.

Stage - 5:

Present: In the final stage, we present a new model with our work on study, analysis, and observations. Our new hybrid model will perform better than existing models and it is described in the next sections.



1. Literature Survey & Define
• Identify and determine academic sources.
2. Publications & Surveys
• Research and learn from recent publications, industriaal surveys and case studies.
3. Technology Adoption
• Identify the technologies adopted in different organizations.
4. Analysis
• Analyse the pros and cons of existing technologies.
• Identify the existing problems.
5. Present
• Present your observation, produce new approach and develop new solutions.

Fig.4. Descriptive Research Methodology employed in our study

- In our work, we have developed a system with two modules, one is to monitor a directory or a file system continuously and alert the admin if there are any changes like file creation, modification, and deletion.
- The other is to compare the hash values of the new suspicious file with our existing file.
- We have used MD5, SHA1, SHA 256, SHA3_512, and CRC 32 hashing algorithms to generate hash values.

Module 1: Directory Monitoring System.

- In the directory monitoring system we used a python library called "Watchdog".
- We developed a simple program to monitor directories as specified in the command-line arguments and generates the event logs.
- This will alert the administrator whenever there are changes in a library.
- If any new suspicious file is detected, then that is sent to the Hash comparison system to compare and verify the file**.**

Module 2: Hash Generator and Verifier.

- In this module we used hashlib, zlib libraries to generate MD5[13], SHA1[14], SHA256[15], SHA3_512[16] hashing algorithms, and CRC32[17] error-detecting code.
- We developed a program to generate the hash value of the given file and compare them with the actual key file.
- If the hash values are matched, then it means that the file is the same then definitely they must have stolen that file. In this way, we can detect data theft.

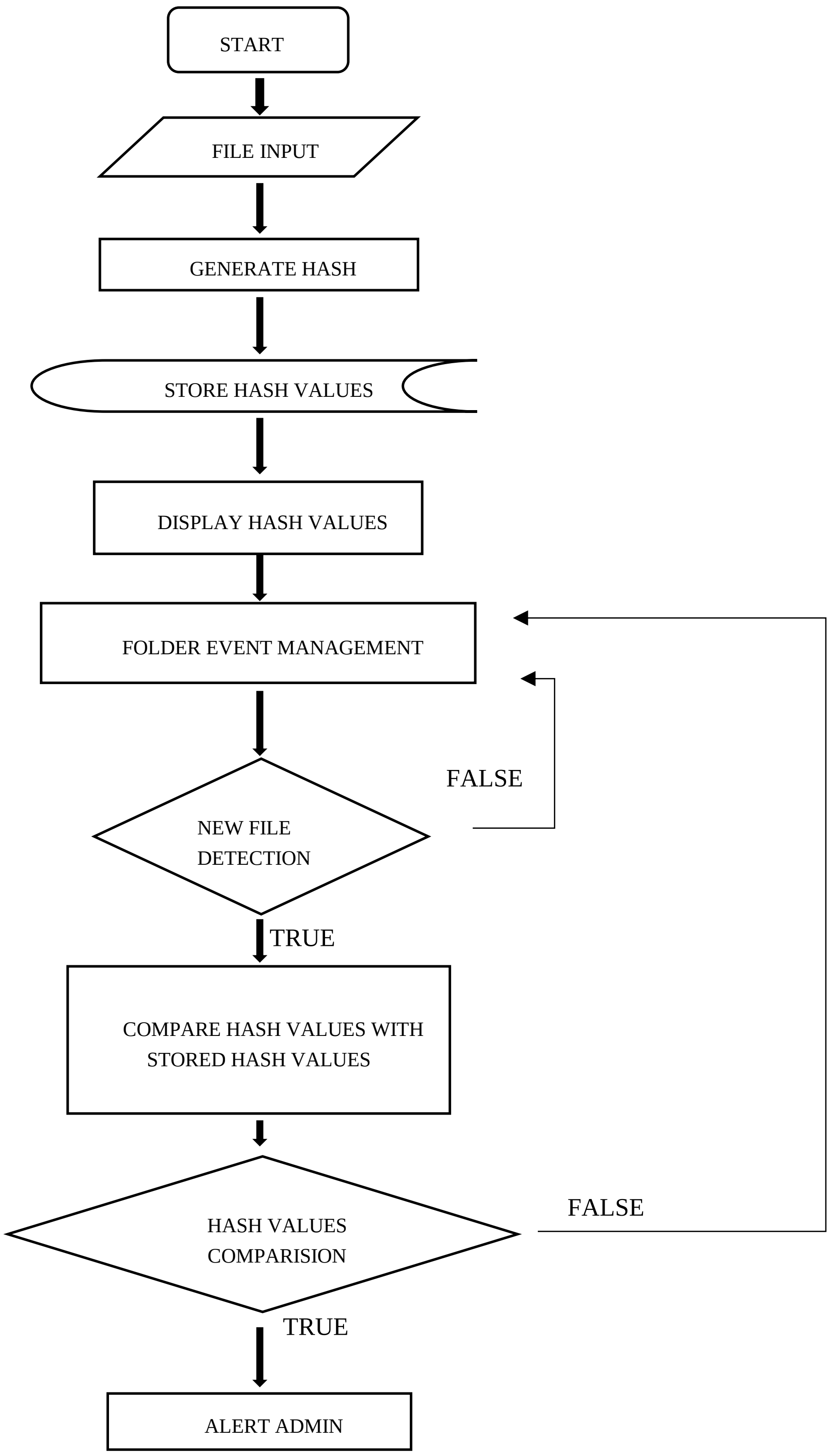


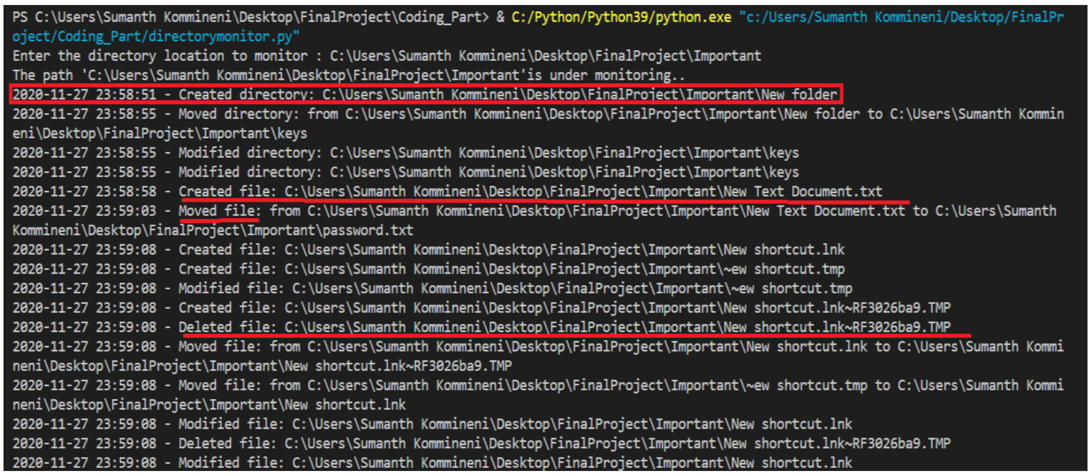
Fig.5. Flowchart of our Insider data theft detection and monitoring system.

- Step 1: Input our Important file.
- Step 2: To generate the hash.
- Step 3: Store & Display the hashes.
- Step 4: If any suspicious file/activity are found in our directory then admin is alerted.
- Step 5: The file is sent for hash verification.
- Step 6: If hash values are matched then Admin is notified else continue monitoring.

4 Results and Discussion

4.1 Results

Output-1:



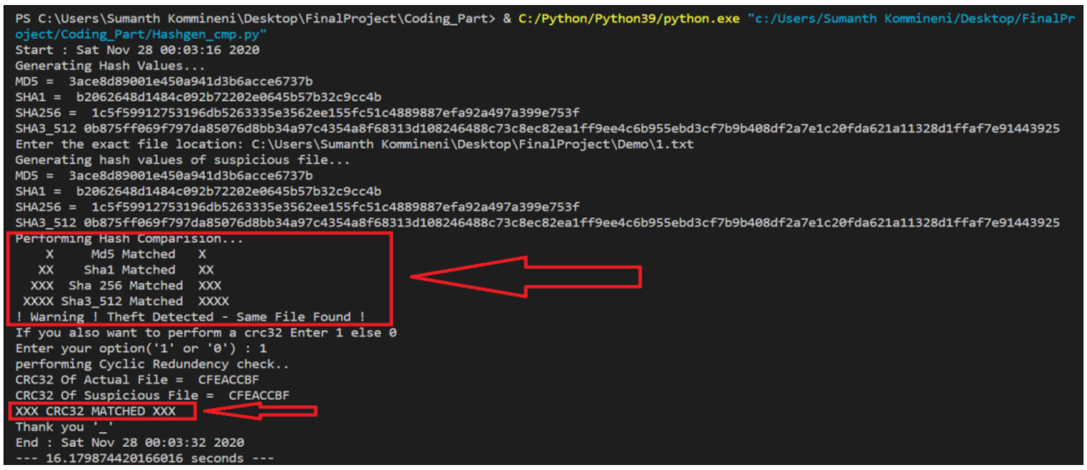
```
PS C:\Users\Sumanth Kommineni\Desktop\FinalProject\Coding_Part> & C:/Python/Python39/python.exe "c:/Users/Sumanth Kommineni/Desktop/FinalPr
oject/Coding_Part/directorymonitor.py"
Enter the directory location to monitor : C:\Users\Sumanth Kommineni\Desktop\FinalProject\Important
The path 'C:\Users\Sumanth Kommineni\Desktop\FinalProject\Important'is under monitoring..
2020-11-27 23:58:51 - Created directory: C:\Users\Sumanth Kommineni\Desktop\FinalProject\Important\New folder
2020-11-27 23:58:55 - Moved directory: from C:\Users\Sumanth Kommineni\Desktop\FinalProject\Important\New folder to C:\Users\Sumanth Kommin
eni\Desktop\FinalProject\Important\keys
2020-11-27 23:58:55 - Modified directory: C:\Users\Sumanth Kommineni\Desktop\FinalProject\Important\keys
2020-11-27 23:58:55 - Modified directory: C:\Users\Sumanth Kommineni\Desktop\FinalProject\Important\keys
2020-11-27 23:58:58 - Created file: C:\Users\Sumanth Kommineni\Desktop\FinalProject\Important\New Text Document.txt
2020-11-27 23:59:03 - Moved file: from C:\Users\Sumanth Kommineni\Desktop\FinalProject\Important\New Text Document.txt to C:\Users\Sumanth
Kommineni\Desktop\FinalProject\Important\password.txt
2020-11-27 23:59:08 - Created file: C:\Users\Sumanth Kommineni\Desktop\FinalProject\Important\New shortcut.lnk
2020-11-27 23:59:08 - Created file: C:\Users\Sumanth Kommineni\Desktop\FinalProject\Important\~ew shortcut.tmp
2020-11-27 23:59:08 - Modified file: C:\Users\Sumanth Kommineni\Desktop\FinalProject\Important\~ew shortcut.tmp
2020-11-27 23:59:08 - Created file: C:\Users\Sumanth Kommineni\Desktop\FinalProject\Important\New shortcut.lnk~RF3026ba9.TMP
2020-11-27 23:59:08 - Deleted file: C:\Users\Sumanth Kommineni\Desktop\FinalProject\Important\New shortcut.lnk~RF3026ba9.TMP
2020-11-27 23:59:08 - Moved file: from C:\Users\Sumanth Kommineni\Desktop\FinalProject\Important\New shortcut.lnk to C:\Users\Sumanth Kommi
neni\Desktop\FinalProject\Important\New shortcut.lnk~RF3026ba9.TMP
2020-11-27 23:59:08 - Moved file: from C:\Users\Sumanth Kommineni\Desktop\FinalProject\Important\~ew shortcut.tmp to C:\Users\Sumanth Kommi
neni\Desktop\FinalProject\Important\New shortcut.lnk
2020-11-27 23:59:08 - Modified file: C:\Users\Sumanth Kommineni\Desktop\FinalProject\Important\New shortcut.lnk
2020-11-27 23:59:08 - Deleted file: C:\Users\Sumanth Kommineni\Desktop\FinalProject\Important\New shortcut.lnk~RF3026ba9.TMP
2020-11-27 23:59:08 - Modified file: C:\Users\Sumanth Kommineni\Desktop\FinalProject\Important\New shortcut.lnk
```

Fig.6. Module 1 Output

Output Explanation:

This is a directory monitoring system; the given directory is monitored and if any event takes place in that directory, then the user is alerted. From the above figure, we can see that in the directory "**C:\Users\Sumanth Kommineni\Desktop\FinalProject\Important\New folder**", a new file "**New Text Document.**txt" is created and the same is shown in the terminal and file has been renamed to "**password.txt**" and other few files were created and deleted which we can see in the terminal. In this way, the Directory monitoring notifies the events in the directory.

Output – 2:



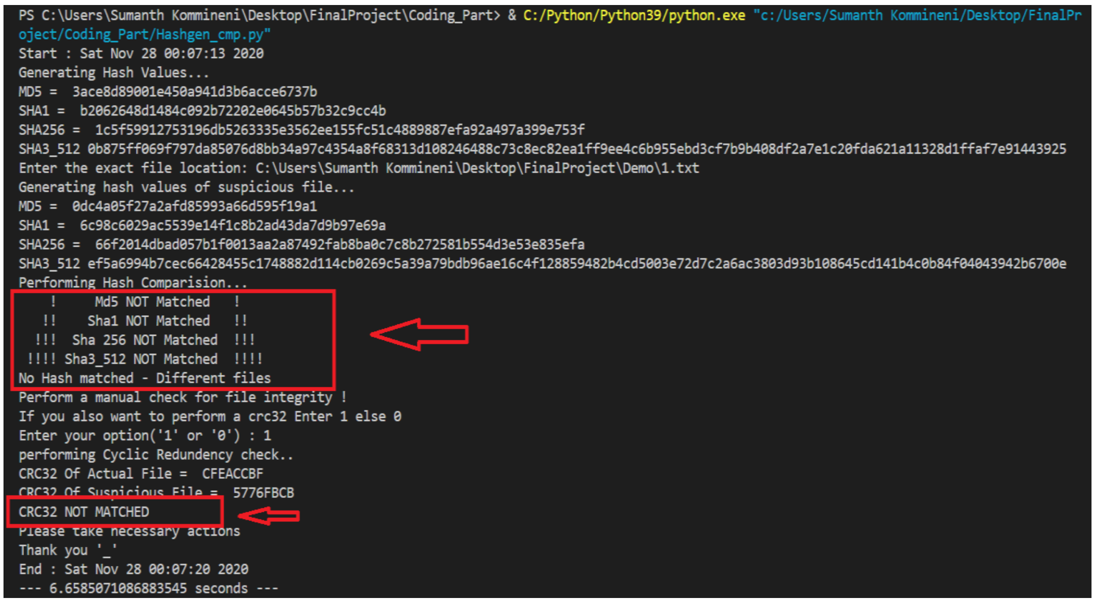
```
PS C:\Users\Sumanth Kommineni\Desktop\FinalProject\Coding_Part> & C:/Python/Python39/python.exe "c:/Users/Sumanth Kommineni/Desktop/FinalPr
oject/Coding_Part/Hashgen_cmp.py"
Start : Sat Nov 28 00:03:16 2020
Generating Hash Values...
MD5 =  3ace8d89001e450a941d3b6acce6737b
SHA1 =  b2062648d1484c092b72202e0645b57b32c9cc4b
SHA256 =  1c5f59912753196db5263335e3562ee155fc51c4889887efa92a497a399e753f
SHA3_512 0b875ff069f797da85076d8bb34a97c4354a8f68313d108246488c73c8ec82ea1ff9ee4c6b955ebd3cf7b9b408df2a7e1c20fda621a11328d1ffaf7e91443925
Enter the exact file location: C:\Users\Sumanth Kommineni\Desktop\FinalProject\Demo\1.txt
Generating hash values of suspicious file...
MD5 =  3ace8d89001e450a941d3b6acce6737b
SHA1 =  b2062648d1484c092b72202e0645b57b32c9cc4b
SHA256 =  1c5f59912753196db5263335e3562ee155fc51c4889887efa92a497a399e753f
SHA3_512 0b875ff069f797da85076d8bb34a97c4354a8f68313d108246488c73c8ec82ea1ff9ee4c6b955ebd3cf7b9b408df2a7e1c20fda621a11328d1ffaf7e91443925
Performing Hash Comparision...
    X     Md5 Matched   X
   XX    Sha1 Matched   XX
  XXX  Sha 256 Matched  XXX
 XXXX Sha3_512 Matched  XXXX
! Warning ! Theft Detected - Same File Found !
If you also want to perform a crc32 Enter 1 else 0
Enter your option('1' or '0') : 1
performing Cyclic Redundency check..
CRC32 Of Actual File =  CFEACCBF
CRC32 Of Suspicious File =  CFEACCBF
XXX CRC32 MATCHED XXX
Thank you '_'
End : Sat Nov 28 00:03:32 2020
--- 16.179874420166016 seconds ---
```

Fig.7. Module 2 Output – True case output

Output Explanation:

A suspicious file named **1.txt**’s location is given for hash comparison. We can see that all the hash values MD5, SHA1, SHA256, SHA3_512 are first generated, printed, and then they are compared. Here all the hash values matched, later CRC32 is also checked, and it also got matched. This means the file is obtained by stealing and the same is prompted there “ **! Warning! Theft Detected – Same File Found !** ” and “ **XXX CRC32 MATCHED XXX**”.

Output - 3:



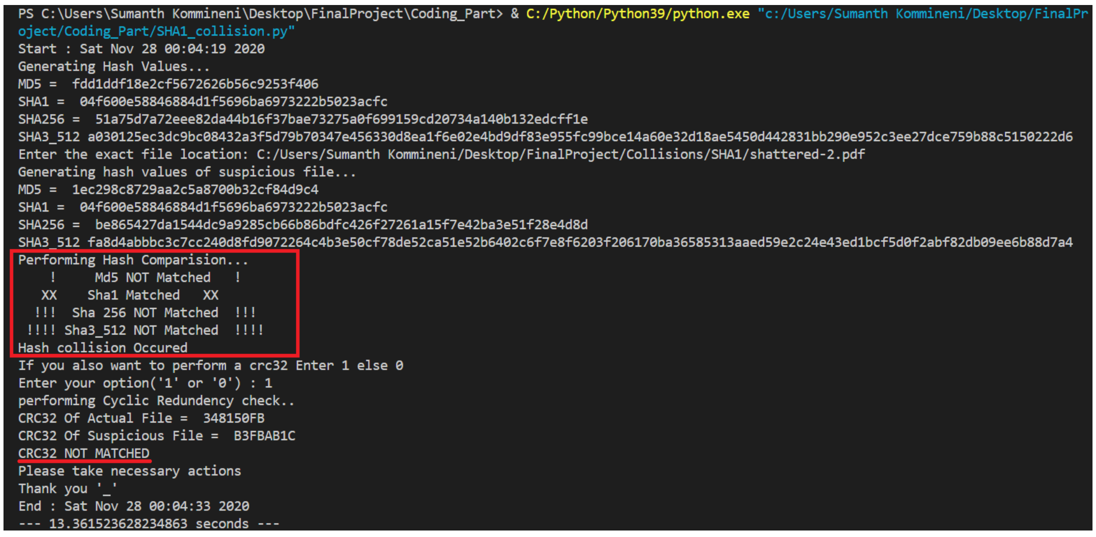
```
PS C:\Users\Sumanth Kommineni\Desktop\FinalProject\Coding_Part> & C:/Python/Python39/python.exe "c:/Users/Sumanth Kommineni/Desktop/FinalPr
oject/Coding_Part/Hashgen_cmp.py"
Start : Sat Nov 28 00:07:13 2020
Generating Hash Values...
MD5 =  3ace8d89001e450a941d3b6acce6737b
SHA1 =  b2062648d1484c092b72202e0645b57b32c9cc4b
SHA256 =  1c5f59912753196db5263335e3562ee155fc51c4889887efa92a497a399e753f
SHA3_512 0b875ff069f797da85076d8bb34a97c4354a8f68313d108246488c73c8ec82ea1ff9ee4c6b955ebd3cf7b9b408df2a7e1c20fda621a11328d1ffaf7e91443925
Enter the exact file location: C:\Users\Sumanth Kommineni\Desktop\FinalProject\Demo\1.txt
Generating hash values of suspicious file...
MD5 =  0dc4a05f27a2afd85993a66d595f19a1
SHA1 =  6c98c6029ac5539e14f1c8b2ad43da7d9b97e69a
SHA256 =  66f2014dbad057b1f0013aa2a87492fab8ba0c7c8b272581b554d3e53e835efa
SHA3_512 ef5a6994b7cec66428455c1748882d114cb0269c5a39a79bdb96ae16c4f128859482b4cd5003e72d7c2a6ac3803d93b108645cd141b4c0b84f04043942b6700e
Performing Hash Comparision...
    !     Md5 NOT Matched   !
   !!     Sha1 NOT Matched   !!
  !!!  Sha 256 NOT Matched  !!!
 !!!! Sha3_512 NOT Matched  !!!!
No Hash matched - Different files
Perform a manual check for file integrity !
If you also want to perform a crc32 Enter 1 else 0
Enter your option('1' or '0') : 1
performing Cyclic Redundency check..
CRC32 Of Actual File =  CFEACCBF
CRC32 Of Suspicious File =  5776FBCB
CRC32 NOT MATCHED
Please take necessary actions
Thank you '_'
End : Sat Nov 28 00:07:20 2020
--- 6.6585071086883545 seconds ---
```

Fig.8. Module 2 – False Case Output

Output Explanation:

A suspicious file named location is given for hash comparison. We can see that all the hash values MD5, SHA1, SHA256, SHA3_512 did not match, later CRC32 is also checked, and it also didn’t match. This means that both are different and the same is prompted there “ **No Hash Matched – Different files** ” and “ **CRC32 NOT MATCHED**”.

Output – 4:



```
PS C:\Users\Sumanth Kommineni\Desktop\FinalProject\Coding_Part> & C:/Python/Python39/python.exe "c:/Users/Sumanth Kommineni/Desktop/FinalPr
oject/Coding_Part/SHA1_collision.py"
Start : Sat Nov 28 00:04:19 2020
Generating Hash Values...
MD5 =  fdd1ddf18e2cf5672626b56c9253f406
SHA1 =  04f600e58846884d1f5696ba6973222b5023acfc
SHA256 =  51a75d7a72eee82da44b16f37bae73275a0f699159cd20734a140b132edcff1e
SHA3_512 a030125ec3dc9bc08432a3f5d79b70347e456330d8ea1f6e02e4bd9df83e955fc99bce14a60e32d18ae5450d442831bb290e952c3ee27dce759b88c5150222d6
Enter the exact file location: C:/Users/Sumanth Kommineni/Desktop/FinalProject/Collisions/SHA1/shattered-2.pdf
Generating hash values of suspicious file...
MD5 =  1ec298c8729aa2c5a8700b32cf84d9c4
SHA1 =  04f600e58846884d1f5696ba6973222b5023acfc
SHA256 =  be865427da1544dc9a9285cb66b86bdfc426f27261a15f7e42ba3e51f28e4d8d
SHA3_512 fa8d4abbbc3c7cc240d8fd9072264c4b3e50cf78de52ca51e52b6402c6f7e8f6203f206170ba36585313aaed59e2c24e43ed1bcf5d0f2abf82db09ee6b88d7a4
Performing Hash Comparision...
    !     Md5 NOT Matched   !
   XX     Sha1 Matched   XX
  !!!  Sha 256 NOT Matched  !!!
 !!!! Sha3_512 NOT Matched  !!!!
Hash collision Occured
If you also want to perform a crc32 Enter 1 else 0
Enter your option('1' or '0') : 1
performing Cyclic Redundency check..
CRC32 Of Actual File =  348150FB
CRC32 Of Suspicious File =  B3FBAB1C
CRC32 NOT MATCHED
Please take necessary actions
Thank you '_'
End : Sat Nov 28 00:04:33 2020
--- 13.361523628234863 seconds ---
```

Fig.9. Module 2 Output – Collision Detection

Output Explanation:

As discussed above, our program is designed in a way to detect hash collisions also. When we execute the module 2 code, then the user is asked to provide the suspicious file path. In the above case, Only SHA – 1 hash value was matching, remaining all didn't match, this is called hash collisions. Even if the files are different, in rare cases same hash values are generated to avoid those we are comparing with 4 hash functions. In this way, hash collisions are detected.

4.2 Limitations and Drawbacks

- The main reason for choosing 5 Hash algorithms is to improve security and integrity.
- We know hash collisions [18] occur in MD5, SHA1 but the chance of occurrence is almost negligence.
- By using 5 algorithms we can detect any theft until the files do not tamper.
- The integrity can be achieved by always encrypting the file or with password protection, Data/Files must be protected with a password or encryption mechanism to avoid data tampering.
- It is mandatory to know what data can be stolen.
- All the servers must be located in DMZ only, the firewall must be designed in a way not to send any important file out of the intranet.
- User Based and Role-Based Profile Assessments should be performed.
- Deploy data loss prevention mechanism and data recovery mechanism.
- Only secured and standard protocols must be used in the data transfer process.
- Employees having access to confidential data should be moved to an isolated area to perform many tasks and have to be monitored round the clock.
- A token-based password system must be implemented so that every time they log in a new session and password are generated. This improves authenticity.
- Reduce the use of applications like G-Mail, Dropbox, Social Media which are 3rd party and potentially vulnerable to our system.
- All the tasks performed must be logged, in IDS, NIDS, HIDS false positives must be examined properly.
- Hash comparison is not a completely reliable system, there might be discrepancies. So, also perform other file theft and insider theft detection mechanisms.
- Insider threats are always costly, so every small mistake costs a penny. So, Operate everything wisely.

5 Conclusion

Data theft is becoming a pivotal theft in IT Industry. Not only in IT, but many institutions, organizations, cloud platforms are also facing it. In our research, we have analyzed the Insider threats and Insider data thefts, how serious they are. We have demonstrated a mechanism to monitor and detect insider data theft. Our program will scan the files, generate the hash values and compare them. It is always a good thing to be secured all the time, attacks may not only happen through outside but also inside. The most important difference between an insider and an outside attacker is about the awareness about the system in which they will be working in an organization. An insider can detour the security infrastructures like IDS and raise an attack vector and exploit from any frightening corner. So, all employees must be given proper guidance and explanation about insider threats, their causes and affect's.

We must be up to date in terms of security to avoid any kind of data breaches and hacks. Further research can be done on our project and project greater and more efficient outcomes. Developments can be done, P – Packet Filtering[6] R – Role-Based and User-Based profile assessment[7] A – Anomaly Detection[19] and F – File Access Logs[20], PRAF methodology can be implemented to obtain greater accuracy and security.

I conclude our research and our knowledge are helpful in terms of minimizing insider data theft and threats.

References:

1. Insider Data Theft: Definition, Common Scenarios, and Prevention, Tips. Available:https://www.ekransystem.com/en/blog/insider-data-theft-definition.
2. 2020 Insider Threat Survey Report by GURUCUL, Available: https://gurucul.com/2020-insider-threat-survey-report
3. 2020 Cost of Insider Threats: Global Report ,Available: https://www.observeit.com/2020costofinsiderthreat/
4. R. A. Alsowail and T. Al-Shehari, "Empirical Detection Techniques of Insider Threat Incidents," in *IEEE Access*, vol. 8, pp. 78385-78402, 2020, doi: 10.1109/ACCESS.2020.2989739
5. N. Patel and K. B. Kansara, "UBM – UVM Approach for Preventing Insider Data Theft from Cloud Storage," *2018 5th International Symposium on Emerging Trends and Technologies in Libraries and Information Services (ETTLIS)*, Noida, 2018, pp. 36-40, doi: 10.1109/ETTLIS.2018.8485217.
6. J. Kim, "Development of Integrated Insider Attack Detection System Using Intelligent Packet Filtering," *2011 First ACIS/JNU International Conference on Computers, Networks, Systems and Industrial Engineering*, Jeju Island, 2011, pp. 65-69, doi: 10.1109/CNSI.2011.4.
7. P. A. Legg, O. Buckley, M. Goldsmith and S. Creese, "Automated Insider Threat Detection System Using User and Role-Based Profile Assessment," in *IEEE Systems Journal*, vol. 11, no. 2, pp. 503-512, June 2017, doi: 10.1109/JSYST.2015.243844
8. R. Kowsik and L. Vignesh, "Mitigating insider data theft attacks in the cloud," 2016 Second International Conference on Science Technology Engineering and Management (ICONSTEM), Chennai, 2016, pp. 561-567, doi: 10.1109/ICONSTEM.2016.7560956.
9. L. Liu, C. Chen, J. Zhang, O. De Vel and Y. Xiang, "Insider Threat Identification Using the Simultaneous Neural Learning of Multi-Source Logs," in IEEE Access, vol. 7, pp. 183162-183176, 2019, doi: 10.1109/ACCESS.2019.2957055.
10. D. C. Le, N. Zincir-Heywood and M. I. Heywood, "Analyzing Data Granularity Levels for Insider Threat Detection Using Machine Learning," in IEEE Transactions on Network and Service Management, vol. 17, no. 1, pp. 30-44, March 2020, doi: 10.1109/TNSM.2020.2967721.
11. Descriptive Research: Definition, Characteristics, Methods, Examples and Advantages. Available: https://www.questionpro.com/blog/descriptive-research/
12. CERT. (2016). Insider Threat Test Dataset. Software Engineering Institute, Carnegie Mellon University. Accessed: May 6, 2018. [Online]. Available: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid= 508099
13. *Rivest, R.* (April 1992). *"Step 4. Process Message in 16-Word Blocks". The MD5 Message-Digest Algorithm*. *IETF*. p. 5. sec. 3.4. *doi:10.17487/RFC1321. RFC 1321. Retrieved 10 October 2018.*
14. Eastlake, D., & Jones, P.E. (2001). US Secure Hash Algorithm 1 (SHA1). *RFC, 3174*, 1-22.
15. A. L. Selvakumar and C. S. Ganadhas, "The Evaluation Report of SHA-256 Crypt Analysis Hash Function," 2009 International Conference on Communication Software and Networks, Macau, 2009, pp. 588-592, doi: 10.1109/ICCSN.2009.50.
16. SHA3-512, Federal Inf. Process. Stds. (NIST FIPS) – 202, https://doi.org/10.6028/NIST.FIPS.202

17. CRC32, Chen, C. & Tian, R. & Yao, Z.. (2013). Design and implementation of the CRC-32 checksum of parallel algorithm. Yi Qi Yi Biao Xue Bao/Chinese Journal of Scientific Instrument. 34. 151-155
18. Xie Tao; Fanbao Liu & Dengguo Feng (2013). "Fast Collision Attack on MD5" (PDF).
19. M. Kim, K. Kim and H. Lee, "Development trend of insider anomaly detection system," 2018 20th International Conference on Advanced Communication Technology (ICACT), Chuncheon-si Gangwon-do, Korea (South), 2018, pp. 373-376, doi: 10.23919/ICACT.2018.8323762.
20. F. Toffalini, I. Homoliak, A. Harilal, A. Binder and M. Ochoa, "Detection of Masqueraders Based on Graph Partitioning of File System Access Events," 2018 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, 2018, pp. 217-227, doi: 10.1109/SPW.2018.00037.