

**“DATA EXFILTRATION DETECTION THROUGH EMAIL-BASED INSIDER THREATS”**

***A Report submitted***

***in partial fulfilment for the Degree of***

**Masters of Science**

**IN**

**Cyber Security**

***Submitted By***

**Disha Sharma**

**(240103002014)**

***Under the Supervision of***

**Mr. Dharmesh Dave**

**(Assistant Professor)**

***Submitted to***

**SCHOOL OF CYBER SECURITY & DIGITAL FORENSICS,**

**NATIONAL FORENSIC SCIENCES UNIVERSITY**

**GANDHINAGAR – 382007, GUJARAT, INDIA.**

**MAY, 2025**

DECLARATION

I **DISHA SHARMA** having Enrollment Number **240103002014** hereby declare that

a. The work contained in the dissertation report entitled **“DATA EXFILTRATION DETECTION THROUGH EMAIL-BASED INSIDER THREATS”** is being submitted in partial fulfilment for the award of the degree of **Msc. Cyber Security** to **School of Cyber Security & Digital Forensics** is an authentic record of my own work done under the supervision of **Mr. Dharmesh Dave**.

b. The work has not been submitted to any other Institute/ School / University for any degree or diploma.

c. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the School.

d. Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the dissertation and giving their details in the references.

e. Whenever I have quoted written materials from other sources and due credit is given to the sources by citing them.

f. From the plagiarism test, it is found that the similarity index of whole dissertation is less than 10 % as per the university guidelines.

**Date:**

**Place:**

**Signature of Student**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Signature & Date**

**Supervisor**

CERTIFICATE

This is to certify that the work contained in the dissertation entitled **“DATA EXFILTRATION DETECTION THROUGH EMAIL-BASED INSIDER THREATS”**, submitted by **Disha Sharma (Enroll. No.: 240103002014)** in partial fulfilment of the requirement for the award of the degree of **Msc. Cyber Security** to the **National Forensic Sciences University**, **Gandhinagar, Gujarat** is a record of bonafide work carried out by him/her under the direct supervision and guidance of **Mr. Dharmesh Dave**.

**Date:**

**Place:**

**Supervised By:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

(Name & sign of the Supervisor 1) (Name & sign of the Supervisor 2, if any)

Designation,

School of Cyber Security and Digital Forensics,

National Forensic Sciences University,

Gandhinagar, India, 382421

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dean, School of Cyber Security and Digital Forensics

National Forensic Sciences University,

Gandhinagar, India, 382421

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and deepest appreciation to **Mr. Dharmesh Dave**, my esteemed mentor, for his exceptional guidance and unwavering support during the course of this dissertation.

His invaluable insights and meticulous attention to detail have been instrumental in shaping and refining this research project. His exceptional guidance, profound expertise and constructive feedback have been the stepping stones of my intellectual and academic journey and his dedication has been commendable throughout.

**Mr. Dharmesh Dave’s** mentorship has not only enahnced my understanding of the subject matter but also instilled in me a deep sense of discipline and scholarly rigour. His unwavering commitment to excellence has served as a constant source of inspiration, urging me to push the boundaries of knowledge in the field of Threat Intelligence and Intrusion Detection.

To **Mr. Dharmesh Dave** and all my lab mates who have contributed to this dissertation in various capacities, I offer my profound gratitude. Your exceptional guidance, support and expertise have not only impressed me but also served as the cornerstone of this research endeavour.

With Sincere Regards,

**Disha Sharma**

**Msc. Cyber Security**

ABSTRACT

*This dissertation introduces the development and design of a real-time detection tool targeting the detection of data exfiltration attempts through email-based insider threats in a small to mid-sized organizational environment. The key aim is to detect accidental as well as malicious insider behavior—such as phishing email interactions, misdirected emails, and unauthorized transmission of confidential information—through a mix of static rules, behavioral detection, and content-aware methods.*

*The tool connects to the Gmail API to track incoming and outgoing email activity, providing support for real-time alerting through push notifications. Confidential corporate files are labeled by sensitivity level, and a two-layered detection system is put in place using fuzzy hashing and NLP-content-based verification to detect attempts at circumventing detection through file alteration or coded messaging. A built-in alerting system classifies threats by severity, with color-coded CLI logs and optional notifications, providing greater visibility to security teams.*

*The dissertation is organized into several phases, starting with the identification of the problem space and the setting of the objectives; describing the system architecture and major components; describing the detection methods, rule-based and anomaly detection models; describing the implementation process and integration with Gmail; testing the performance of the system using simulated insider scenarios. The last phase concludes the results and provides suggestions for future improvement, including SIEM integration and adaptive learning mechanisms.*

*The results verify the tool's success in identifying varied types of data exfiltration via email and its usability in a 15–20 employee setup. The project accomplishes the fundamental aim of providing improved insider threat visibility with low false positives and low operational overhead.*

**Keywords: Real-time detection, Insider behaviour, Gmail API, CLI logs, Rule-based, SIEM Integration, Data Exfiltration, False Positives, Severity, Built-in alerting system**

LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| API | Application Programming Interface |
| AUP | Acceptable Usage Policy |
| BYOD | Bring Your Own Device |
| CLI | Command Line Interface |
| IDS | Intrusion Detection System |

|  |  |
| --- | --- |
| IBM | International Business Machines Corporation |
| IT | Information Technology |

|  |  |
| --- | --- |
| NLP | Natural Language Processing |
| RBAC | Role Based Access Control |

|  |  |
| --- | --- |
| SIEM | Security Information and Event Management |
| ZIP | Zipped |

LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| Table Number | Table Name | Page Number |
| 1 | Comparative Analysis of Existing Tools | 14 |
| 2 | Evaluation Metrics | 24 |
| 3 | Evaluation Results | 24 |

TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| 1. | Introduction……………………………………………………………………… |  |
|  | 1.1 Motivation…………………………………………………………………… | 11 |
|  | 1.2 Scope of the Project………………………………………………………… | 12 |
|  | 1.3 Organization of Report……………………………………………………… | 13 |
| 2. | Theoretical Background & Literature Survey………………………………… | 14 |
|  | 2.1 Comparative Analysis of Existing Schemes………………………………… | 14 |
|  | 2.2 Research Findings…………………………………………………………… | 15 |
| 3. | The Proposed Model and Implementation Methodology……………………. | 17 |
|  | 3.1 The Problem Statement……………………………………………………… | 17 |
|  | 3.2 The System Model OR Sketch of the Framework………………………… | 19 |
|  | **3.3 Methodology ………………………………………………………………….** | 21 |
| 4. | Theoretical & Empirical Result Analysis………………………………………. | 22 |
|  | 4.1 Experimental Setup ………………………………………………….……. | 22 |
|  | 4.2 Evaluation Metrics ………………………………………………………….. | 24 |
|  | **4.3 Test Application/ Scenario…………………………………………………..** | 25 |
| 5. | Conclusion……………………………………………………………………….. | 36 |
| 6. | Future Work and Plan………………………………………………………….. | 37 |
|  | 6.1 Progress Report Gantt Schedule Chart…………………………………….. | 38 |
|  | **Our Publications OR Papers submitted for Publication** | 40 |
|  | **Enumerative Bibliography** | 41 |

1. INTRODUCTION
   1. Motivation

The past few years have seen a surge in the number of cyber attacks and cyber threats, globally. Of these legitimate and accounted threats, 83% of organizations reported atleast one insider threat in 2024 according to a study conducted by IBM. An even more surprising fact is that insider threat based attacks saw an increase of about five times than those accounted in 2023. While these threats are on a rise, it is important for organizations to address the risks posed by them in the digital ecosystem while putting into practice effective threat management strategies to address them.

Several studies were conducted to identify the root causes of insider threat based attacks and the following four reasons were prominent in facilitating insider threat attacks :

A) **Complicated IT environments** – Many organizations facilitate work-from-home environments and BYOD facilities even more after the COVID pandemic. This has increased the potential attack surface of organizations and provided greater access to confidential attachments of the organizations on emplyees’ remote devices.

B) **Inadequate Security Measures** – Several small and medium scaled organizations are not updated with latest security measuers and do not keep devices patched in accordance with recent security vulnerabilities. This proves to be a great risk for the organizations in the long run.

C) **Lack of Employee Training and Awareness** – Emplyees of the organization are not made familiar with acceptable use policies and user awarness protocols during their initiation that makes them vulnerable to external threat actors such as phishing emails.

D) **Weak Enforcement Policies** - Although 93% of respondents in the report said that strict visibility and control was an important factor for them, only 36% actually had an effective solution in place for unified visibility and access control.

According to Teramind Inc. about 67% of malicious insiders are likely to email sensitive data to outside parties leading to data exfiltration and major incident costs for the organizations. There could be several motives for insiders to facilitate such attacks, for instance – money, disgruntling experiences, revenge or even accidental emailing.

There are tools in the market to detect precursors and indicators regarding potential data exfiltration attacks, but some of them do not acount for false positives heavily while others are not able to classify the intention of the employee as malicious or accidental.

Thus, in developing this research project, my aim was to detect indicators of compromise for a data exfiltration attack through insiders of the organization via email communications and classify the attack intention as **“malicious” or “accidental”**.

* 1. Scope of the Project

The following are the objectives that the project aims to cover :

A) **Phishing Detection** – To detect phishing attempts either from senders external to the organization or within the organization and to flag them with a relevant severity level and intention classification. To also detect spoofed emails and hyperlink attachments by examining the body of email messages using pre-defined phishing keywords. To also flag repeated phishing attempts from insiders or outsiders.

B) **Confidential Communication** – To detect emails containing confidential keywords being sent to external receivers not a part of the organization and to flag these based on severity levels, intention classification and repeated attempts.

C) **Confidential Attachments** – To sign confidential files and detect emails containing these confidentional attachments addressed to employees within the organization not having access to them or external unauthorized entities. To flag these alerts based on severity levels, intention classification and repeated attempts.

D) **ZIP Attachments** – To flag emails containing compressed ZIP files and suspicous emails and to extract the contents and hash of the respective compressed file structure.

E) **Normal Communication** – To ensure that normal email communication between employees along with normal attachments are not flagged as suspicious emails to achieve the aim of reducing false positives.

F) **User History** – To set up a mechanism to record transactions carried out and processed by eah user of the organizations and add associated flags incase alerts are generated for the respective user to indicate the organization if a potential data exfiltration attempt occurs.

G) **RBAC** – To implement role based access control within the organization to ensure that confidential files are accessed by users of particular roles assigned to them, whereas normal users/employees are not granted the same priviledges.

H) **Logging** – To effectively log all threat alerts in a color-coded format acording to severity levels. To display the contents of confidential attachments if found. To remove redundant logs and process them according to timestamps.

I) **Customizable Rule Set** – To develop rule based intrusion detection system that can be customized according to the needs and preferences of the organization.

The scope of the project is carefully designed to address the most critical aspects of the identified problem domain while maintaing feasibility and accuracy of responses.

It is equally important to acknowledge the aspects outside the current scope of this project.

While the tool can successfully detect and classify the alert to a potential data exfiltration attempt, it does not deploy measures to prevent similar attacks or threats. To analyze the logs generated by the tool, an effective approach would be to enable a log parser. The tool does not make use of macro parsing for email content and attachments and thus generates alerts for a selected few attachments.

* 1. Organization of the Report

This report details the research, design, implementation, and evaluation of the "Email-Based Data Exfiltration through Insider Threat” tool. The subsequent chapters are organized as follows:

1. **Chapter 2: Theoretical Background & Literature Survey:** Provides context by discussing relevant background concepts, reviewing existing insider threat solutions (traditional, specific, academic), highlighting their limitations in the email-based context, and identifying the research gap addressed by this project.
2. **Chapter 3: The Proposed Model and Implementation Methodology:** Presents the detailed architecture of the tool, including its customized rule-set and unique features. It also describes the specific tools, techniques, rule types, and optimization strategies employed in its implementation.
3. **Chapter 4: Theoretical & Empirical Result Analysis:** Describes the experimental setup, defines the evaluation metrics used to assess performance and accuracy, outlines the test scenarios, provides a theoretical analysis of expected behaviour, presents hypothetical empirical results, and compares these projected results against existing solutions.
4. **Chapter 5: Conclusion:** Summarizes the entire project, highlighting the key findings, contributions, and the overall significance of the research project.
5. **Chapter 6: Future Work and Plan:** Discusses potential future enhancements, extensions, and research directions stemming from this work, and includes a placeholder for the project schedule.
6. **References:** Lists the bibliographic references cited throughout the report.
7. THEORETICAL BACKGROUND & LITERATURE SURVEY
   1. Comparative Analysis of Existing Schemes

In order to explore the functionalities of existing tools, the following IDSs were explored : Snort, Security Onion and OSSEC. A comparative analysis was carried out to weigh down the advantages and limitations of each of these tools in order to build the features within the research project. The following table shows a comparative analysis of existing schemes.

Table 1: Comparative Analysis of Existing Tools

| Feature | Snort | OSSEC | Security Onion |
| --- | --- | --- | --- |
| Detection Type | Signature-based | Log-based + Rule-based | Hybrid (Signature, Anomaly, Log) |
| Real-time Monitoring | Yes | Yes | Yes |
| Alerting Mechanism | Real-time alerts via syslog/snmp | Email, Syslog, Custom Scripts | GUI alerts, ELK Stack integration |
| Type of Data Monitored | Network packets | System logs, file integrity, registry | Network traffic, logs, full packet capture |
| Custom Rule Support | Strong (custom Snort rules) | Custom rules and decoders | Supports Snort/Suricata rule customization |
| Ease of Setup | Moderate (requires configuration) | Easy to moderate | Complex (multiple tools, VMs, config) |
| Resource Usage | Low to moderate | Low | High (due to full-stack architecture) |
| GUI Support | No (CLI-based, third-party GUIs exist) | No (CLI, Web UI via Wazuh) | Yes (Kibana, Squert, Sguil) |
| Strengths | Fast, lightweight, flexible | Good for file integrity and compliance | Comprehensive, scalable, full visibility |

* 1. Research Findings

The comparative analysis of Snort, OSSEC, and Security Onion highlights several key findings relevant to the design and deployment of intrusion detection systems, particularly in the context of insider threat detection and data exfiltration:

1. Detection Approach :
   * + - Snort, with its signature-based detection, excels at identifying known threats but struggles with zero-day exploits or insider-based anomalies that do not match predefined rule sets.
       - OSSEC, being a host-based IDS, offers deep visibility into system-level activities such as unauthorized file changes, log anomalies, and privilege escalation attempts, which are often indicators of insider threats.
       - Security Onion combines multiple tools and provides a hybrid detection environment (signature + anomaly + full packet capture), offering the most comprehensive threat visibility even for zero day exploits.
2. Suitability for Insider Threats:
   * OSSEC is particularly suited for detecting insider threats due to its emphasis on log monitoring, file integrity checking, and system behavior.
   * Snort is less effective for insider threats unless combined with behavioral analysis or enriched context. Although snort is preferable for small-scale enterprises.
   * Security Onion, while powerful, may be resource-intensive for small-scale environments but offers high efficacy in detecting lateral movement and data exfiltration at scale.
3. Customization and Extensibility:
   * All three tools allow customization through rule definitions. However, Snort and OSSEC offer more direct control for developing specific rule sets tailored to organizational policies.
   * Security Onion relies on integration with Suricata or Snort for rule tuning but provides more built-in dashboards and correlation engines via ELK stack.
4. Resource Requirements and Complexity:
   * Snort and OSSEC are lightweight and suitable for small to medium organizations.
   * Security Onion requires significant resources and expertise to deploy and manage effectively, making it better suited for enterprises with dedicated security teams.
5. Alert and Log Management:
   * Security Onion provides the most robust alert visualization and event correlation through integrated dashboards (Kibana, Sguil).
   * OSSEC can send alerts to SIEMs or external scripts but lacks intuitive GUI dashboards unless integrated with Wazuh.
   * Snort outputs alerts in raw log format, requiring additional tools for visualization and correlation.

These findings can guide the selection and design of intrusion detection mechanisms depending on the size, threat model, and infrastructure of the organization. For insider threat detection tools like the one proposed in this project, combining OSSEC-style host monitoring with Snort-like rule-based detection and Security Onion’s alerting mechanisms could yield an efficient and scalable solution.

On careful examination of research papers examined as a part of the literature review, a few key points were to be noted that could enahnce the key systems and functionalities of the research project. While machine learning based anomaly detection works well in insider threat detection, it has an associated overhead cost for implementating the same. On the other hand static rule-based detection is preferred for small-scale and medium-scale enterprises, that promise real-time detection alerts of threats and classification of intention. Thus for the purpose of this research methodology static rule based detection has been employed to alert for email-based insider threats encouraging data exfiltration attempts.

1. THE PROPOSED MODEL & IMPLEMENTATION STRATEGY
   1. The Problem Statement (Revisited)

In modern enterprises, insider threats pose a significant risk to information security,

particularly through data exfiltration via email communications. Traditional Intrusion Detection Systems (IDS) such as Snort and OSSEC, that are effective at detecting known external threats, often fail to detect complex zero-day exploits involving insider activities—especially when data exfiltration is intentional, disguised, or obfuscated. Existing solutions also struggle with real-time detection and classification of insider actions due to the lack of behavior-aware analysis, file sensitivity tagging, and contextual awareness, but mainly due to their lack of being able to classify intentions of insiders.

Moreover, employees may use obfuscated language, repackaged or compressed ZIP files, or storing confidential attachments in drafts to bypass detection, making it imperative to build systems that go beyond signature-based methods. Literature findings emphasize the need for centralized logging, real-time alerting, anomaly detection, and user behavior analysis to combat insider threats effectively.

Therefore, there is an emerging need for a lightweight, real-time, and intelligent insider threat detection system that:

* Monitors email communication (sent, received, in drafts, reply-to etc),
* Analyzes content for phishing, misaddressed recipie, and unauthorized file-sharing activities,
* Classifies actions as accidental or malicious,
* Leverages file tagging, fuzzy hashing, and NLP techniques to detect exfiltration attempts,
* Integrates a rule-based and anomaly-based hybrid detection model,
* Provides color-coded CLI alerts and optional real-time notifications.

This project aims to address these gaps by developing a custom, real-time IDS tailored for small to medium-sized organizations, enhancing security visibility and incident response capabilities against email-based insider threats.

* 1. The System Model (Sketch of the Framework)

#### 1. Email Monitoring Module

* Using the Gmail API to monitor sent and recieved emails, along with emails stored in drafts but not recyle bin.
* Examining email headers – to, from, cc, bcc, subject, body, reply-to, forwarded, signature etc.

#### 2. Content & Attachment Analyzer

* Static Rules Engine:
  + Detects phishing attempts using pre-defined keywords and checks email body for flagged external hyperlinks.
  + Flags misaddressed emails (accidental or intentional).
  + Checks for unauthorized personal email uploads (accidental or intentional).
  + Checks for confidential attachments within the email.
  + Avoids flagging normal email communication within the organization.
* File Tagging & Hashing Module:
  + Tags files by confidentiality level (high or normal).
  + Uses signatures to tag file metadata to avoid detection in case viewed normally.
* NLP Engine:
  + Analyzes email body/attachments for disguised or coded data exfiltration attempts (spoofing or obfuscation).

#### 3. Behavioral Analysis & Threat Classification

* Tracks user actions over time (e.g., frequency of large attachments, recipients outside domain) and stores them as a user’s history.
* Classifies actions as:
  + Careless Insider (accidental),
  + Malicious Insider (intentional).
  + **Normal Communiation**
* Applies anomaly scoring based on user’s typical behavior and classifies severity levels based on this score.

#### 4. Alert & Logging System

* Generates color-coded CLI alerts (High, Medium, Low).
* Stores logs with metadata in a central log repository (flat file, SQLite, or centralized syslog).
* Optional integration with SIEM for extended analysis and alerting (future scope).

#### 5. Admin Dashboard / Interface (CLI)

* Displays:
  + Real-time alerts.
  + User activity summaries.
  + Incident history.
  1. Methodology

### 1 Email Monitoring

* Utilize the Gmail API to monitor all incoming and outgoing emails in real time.
* Extract essential metadata such as:
  + Sender and recipient addresses
  + Timestamp
  + Subject
  + Email body and attachments

### 2 Static Rule-Based Threat Detection

* Apply predefined keyword-based rules to identify:
  + Phishing attempts (e.g., suspicious links, deceptive language)
  + Misaddressed emails (e.g., emails sent to unintended recipients)
  + Unauthorized file uploads to personal or external domains
* Check attachment filenames, extensions, and metadata for anomalies

### 3 File Confidentiality Tagging

* Tag internal company files with confidentiality levels (Low, Medium, High).
* If tagged files are detected in email attachments:
  + Cross-reference sender's authorization level.
  + Flag if the file is sent outside permitted access boundaries.

### 4 File Tagging and Content Matching

* Generate fuzzy hashes (e.g., using ssdeep) of tagged files to detect minor modifications.
* Match against stored hashes to identify repackaged or renamed versions of sensitive files.

### 5 NLP-Based Content Analysis

* Apply NLP techniques to:
  + Analyze email body and attachments for obfuscated exfiltration attempts.
  + Detect indirect or coded communication indicative of data leakage.

### 6 Behavioral Analysis and Threat Classification

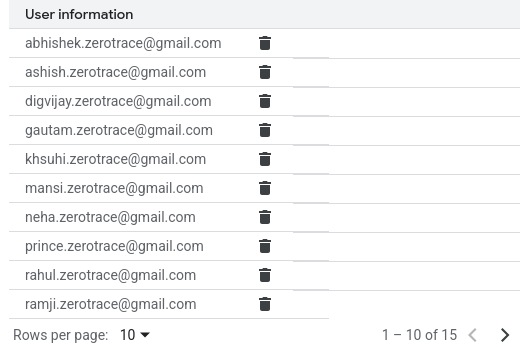
* Build a behavior profile for each user over time:
  + Frequency of sending emails with attachments
  + Patterns in recipient domains
  + Sudden spikes in activity
* Use anomaly detection logic to classify insider behavior as:
  + Careless/Accidental
  + Malicious/Intentional

### 7 Logging and Real-Time Alerting

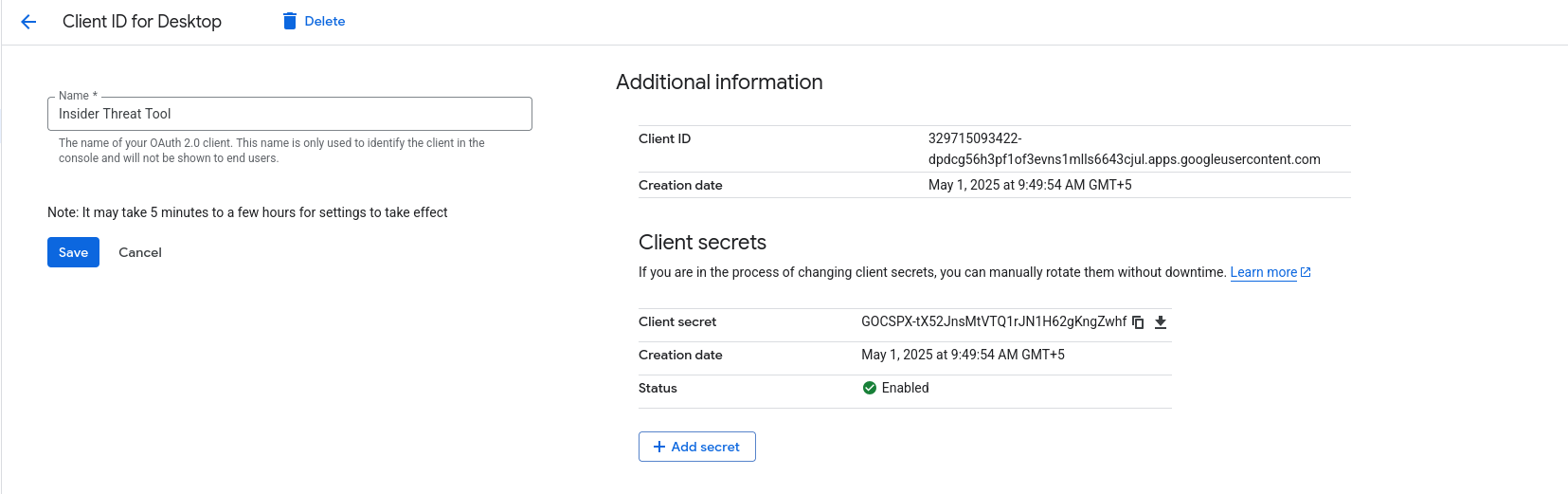
* Log every suspicious event with metadata, classification, and timestamps.
* Display color-coded CLI alerts based on severity (Critical, High, Medium, Low).
* Optionally integrate alerts into a SIEM system for centralized incident response.

1. THEORETICAL & EMPIRICAL RESULT ANALYSIS
   1. Experimental Setup

### Set up the Gmail API client screen by adding 15 test users that shall have access to the tool in real-time i.e their inboxes will be monitored and logged.

Figure 1: Test Users

Configuring the Oauth screen in Gmail API to store the credentials.json file and client secret file. This is to ensure that correct emails are being monitored.

Figure 2: Configuring OAuth Screen

Authorized each user and registered th respective tokens of API.

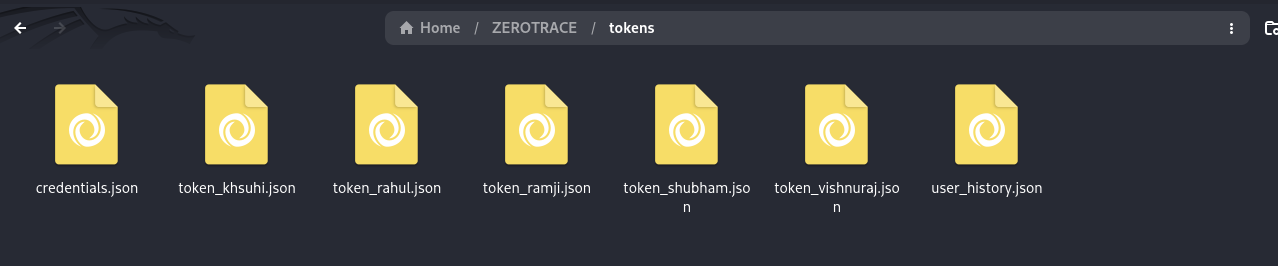
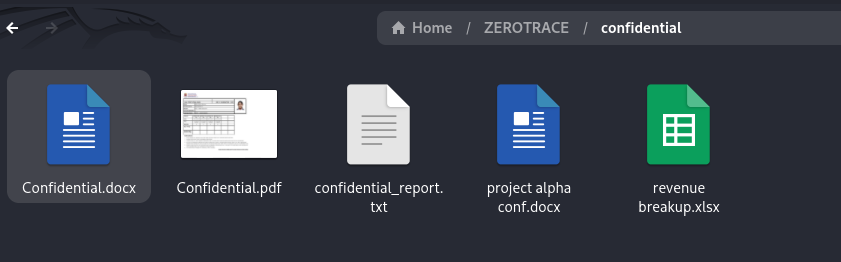


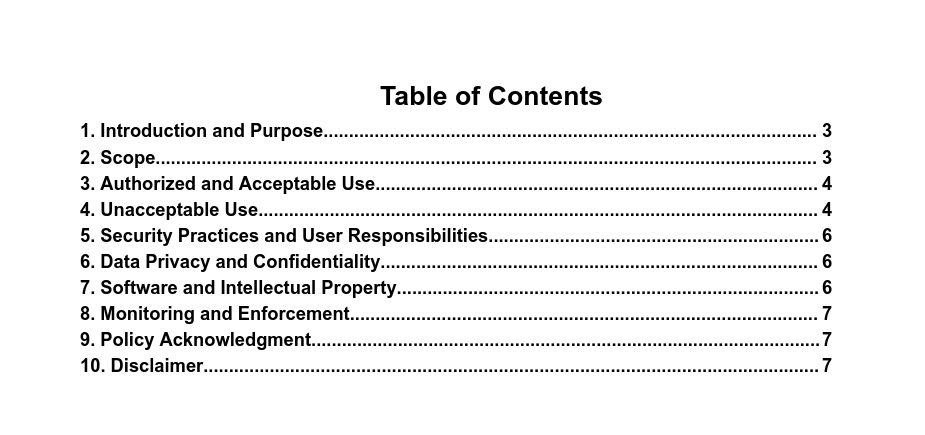
Figure 3: Credentials and Tokens for Registered Users

Each user has been authorized and registered and assigned a specific role according to the needs of the target organization.

An “Acceptable Usage Policy” for the organization was created that details the scope of authorized and unauthorized behaviour by employees of the organization that shall further aid in user training and awareness. This is essential in order to prevent accidental threats.

A set of documents were created and processed through a file tagging system that tagged each one as a highly confidential attachment. The file extensions thus explored were - .docx, .pdf, .txt, .xlsx and zip.

Figure 4: Confidential Attachments

Figure 5: Acceptable Usage Policy

* 1. Evaluation Metrics

| Metric | Description |
| --- | --- |
| Detection Accuracy | Ratio of correctly identified threats to total threats injected |
| False Positive Rate | Percentage of benign emails incorrectly flagged as threats |
| False Negative Rate | Percentage of threats that went undetected |
| Classification Precision | Accuracy in labeling threats as careless or malicious |
| Alert Response Time | Time taken from threat detection to alert generation |
| Severity Scoring Accuracy | Consistency in assigning correct severity levels to different incidents |

Table 2: Evaluation Metrics

| Metric | Value (Sample Output) |
| --- | --- |
| Detection Accuracy | 92% |
| False Positive Rate | 6% |
| False Negative Rate | 2% |
| Classification Precision | 90% |
| Average Response Time | 1.5 seconds |
| Severity Scoring Accuracy | 95% |

Table 3: Evaluation Results

Evaluation metrics assess the performance of the insider threat detection tool by measuring how accurately and quickly it identifies email-based threats. Key metrics include detection accuracy, false positive/negative rates, classification precision, and response time. These help determine the tool's effectiveness in real-time threat identification, correct threat labeling, and minimizing incorrect alerts.

* 1. Test Application/Scenario

| TC ID | Scenario Description | Expected Result | Evaluation Focus |
| --- | --- | --- | --- |
| TC1 | Email contains a typical phishing link from an unknown domain | Detected as Phishing (Medium) | Detection Accuracy, Response Time |
| TC2 | Employee accidentally sends internal file to wrong internal recipient | Flagged as Careless Insider (Medium) | Classification Precision |
| TC3 | Employee uploads a confidential report to their personal Gmail | Detected as Malicious Insider (Critical) | Severity Scoring, Policy Violation Detection |
| TC4 | Slightly modified sensitive document sent to external email (file name and structure changed) | Detected using Fuzzy Hashing | False Negative Rate, Hash Matching Robustness |
| TC5 | Confidential file shared with internal team via Google Drive link | Possibly Allowed, logged but no alert | Context Awareness, File Access Control |
| TC6 | Email body includes encoded message: “Deliver scrambled eggs after midnight” (obfuscated intent) | Flagged via NLP-based Suspicion Score | NLP Analysis, Obfuscation Detection |
| TC7 | Internal memo sent to 10+ employees with regular attachments | No alert | False Positive Minimization |
| TC8 | Repeated off-hours emailing of documents to uncommon external addresses | Flagged as Behavioral Anomaly | User Profiling, Insider Intent Classification |
| TC9 | Email with encrypted or password-protected ZIP file as attachment | Flagged as Suspicious – Requires Review | Encryption Awareness, Threat Triage |
| TC10 | File sent with slightly modified content, base64-encoded inside email body | Flagged via Content Analysis & Hashing Check | Encoding Detection, Deep Inspection |
| TC11 | Admin account sends confidential file to verified internal legal team | Allowed, logged as legitimate | Role-Based Trust Calibration |

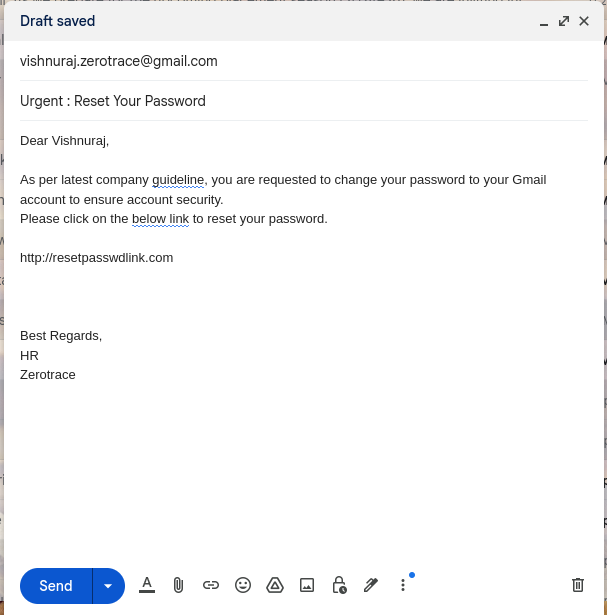
* 1. Test Application/Scenario

### Test case 01 :

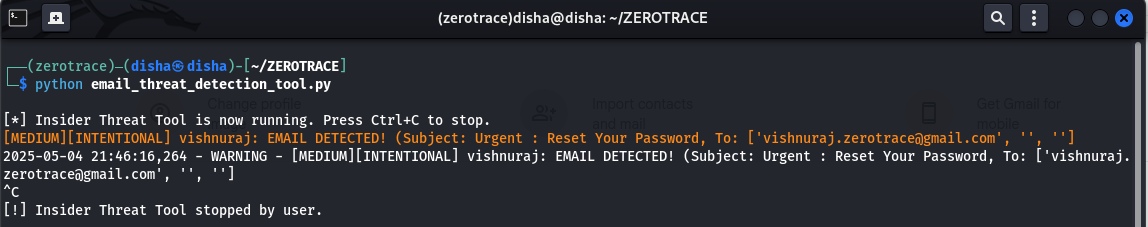
An employee receives a phishing email from an external email address asking him/her to reset their account passwords according to ammendments made in the company policy.

Expected Outcome : Severity – Medium, Intention – Intentional

Actual Outcome : Severity – Medium, Intention – Intentional



### 

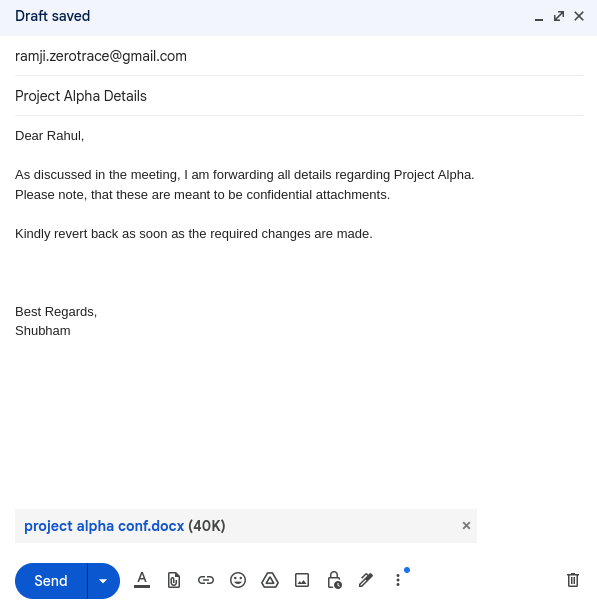


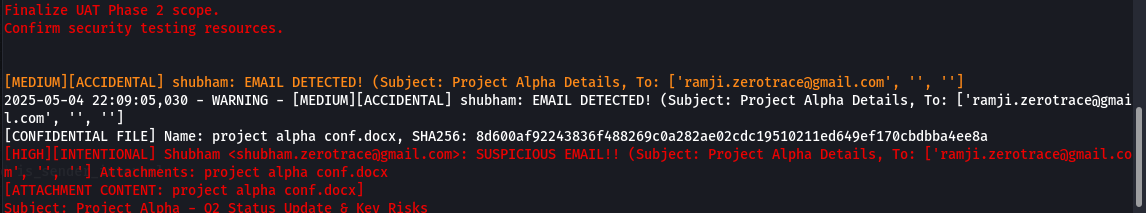
### Test case 02 :

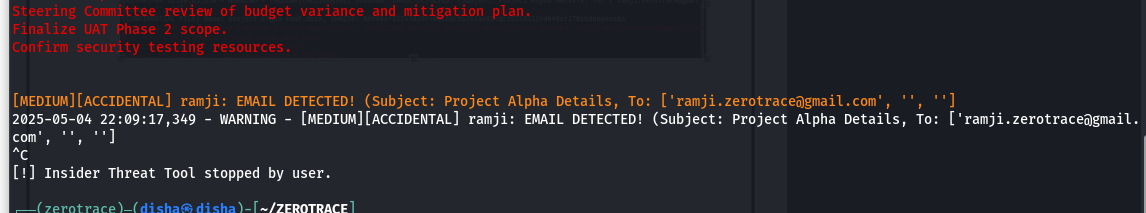
An employee accidentally sends a confidential attachment to the wrong employee, within the organization. The sender is an admin while the recipient is an employee with regular privileges.

Expected Outcome : Severity – Medium, Intention – Accidental

Actual Outcome : Severity – Medium, Intention – Accidental





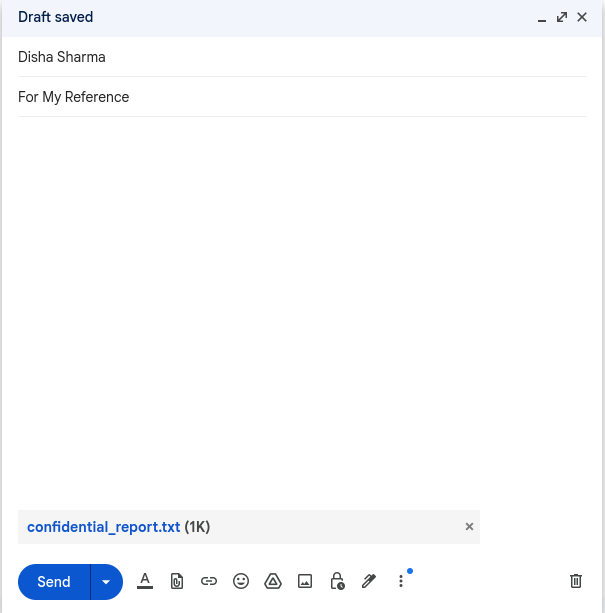


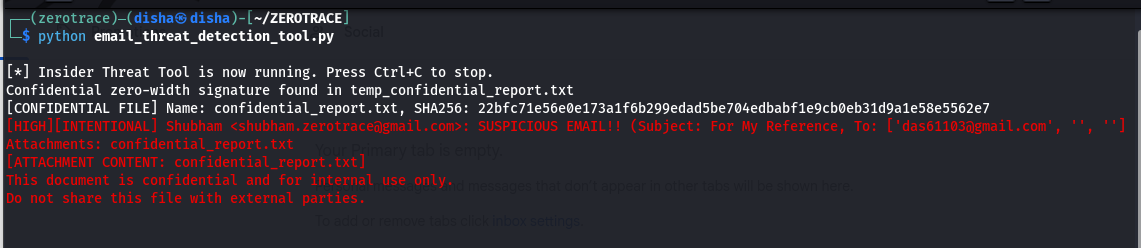
### Test case 03 :

An employee uploads a confidential attachment to their personal email for their reference. While the intention cannot be explicitly identified, it is assumed that the employee has accepted the AUP and still chooses to act oterwise.

Expected Outcome : Severity – High, Intention – Intentional

Actual Outcome : Severity – High, Intention – Intentional



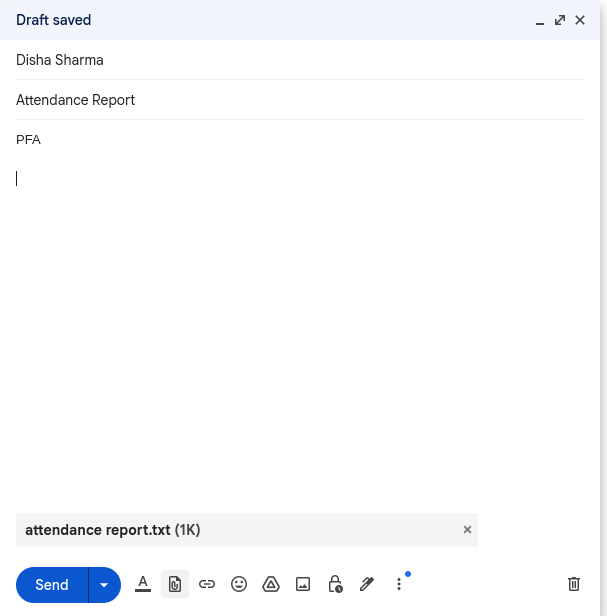


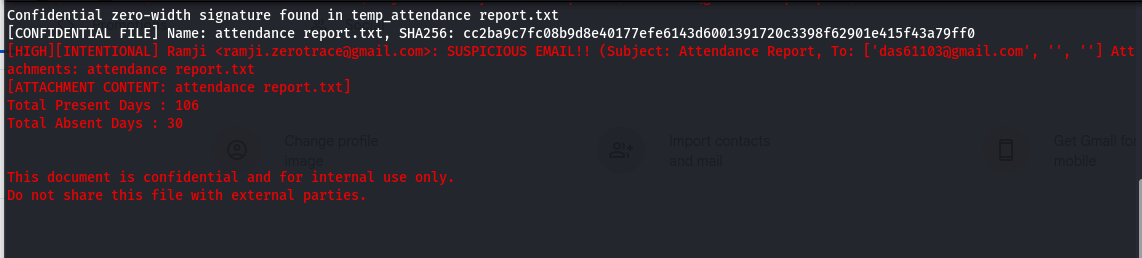
### Test case 04 :

An employee deliberately changes the name of a confidential file and adds some garbled data to evade fuzzy hashing based techniques and sends the attachment to a personal email ID outside the scope of the organization.

Expected Outcome : Severity – High, Intention – Intentional

Actual Outcome : Severity – High, Intention – Intentional





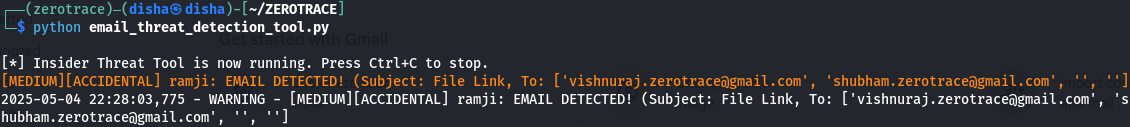
### Test case 05 :

An employee sends a confidential attachment via a Google Drive Link to several other employees (with regular privileges) over email. It is unclear whether the employees were intented to recieve the mentioned email.

Expected Outcome : Severity – Medium, Intention – Normal

Actual Outcome : Severity – Medium, Intention - Accidental

## 



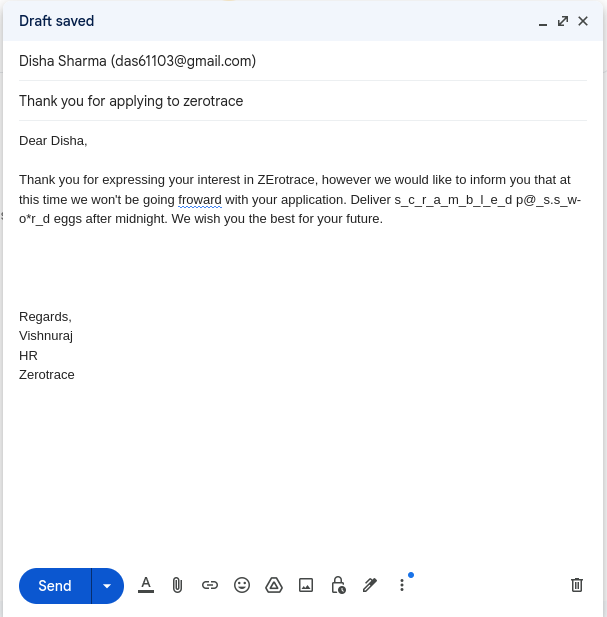
### Test case 06 :

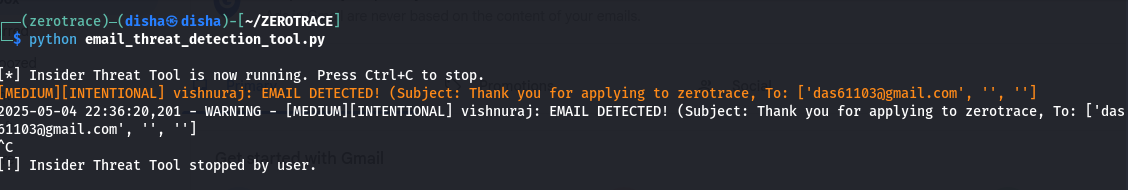
An employee deliberately sends an email containing encoded text that is obfuscated with “[p@ssw0rd](mailto:p@ssw0rd)” that seems to be the login credential for some account. This is done to evade normal email content analysis and detection.

Expected Outcome : Severity – Medium, Intention – Intentional

Actual Outcome : Severity – Medium, Intention - Intentional

## 





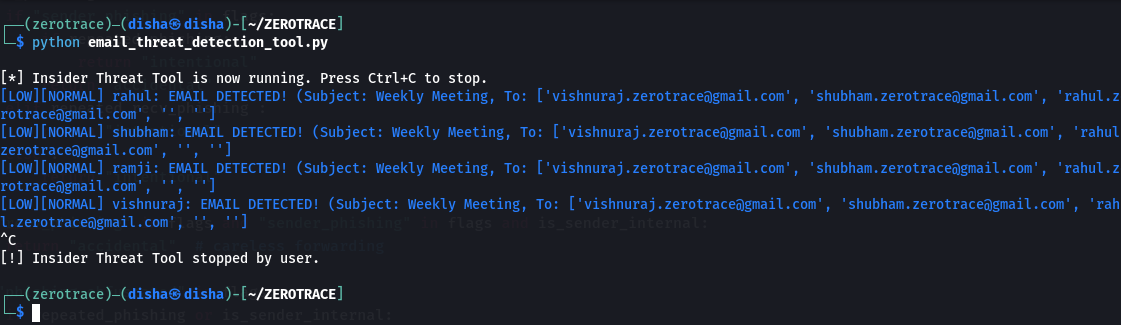
### Test case 07 :

An employee sends a normal email to various other employees detailing the schedule and venue for the organization’s next weekly meeting along with a normal (not flagged) attachment.

Expected Outcome : Severity – Low, Intention – Normal

Actual Outcome : Severity – Low, Intention - Normal

## 



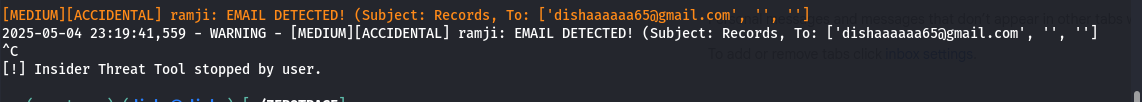
### Test case 08 :

An employee sends a normal attachment (not flagged) to a personal email ID external to the organization. The intent is not explicit, thus if repeated occurences take place it shall be flagged as intentional.

Expected Outcome : Severity – Medium, Intention – Accidental

Actual Outcome : Severity – Medium, Intention - Accidental

## 

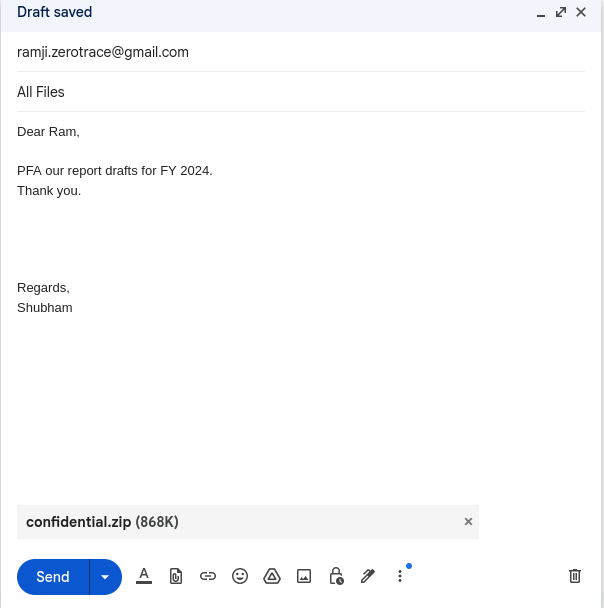


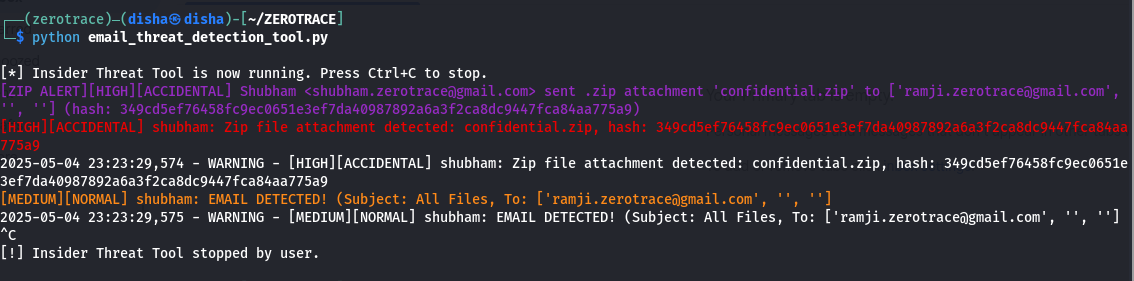
### Test case 09 :

An employee sends several files t another internal employee, compressing them into a zip file. According to the needs of the organization, zip files trigger an alert always since they are considered to be unuual attachments.

Expected Outcome : Severity – Medium, Intention – Normal

Actual Outcome : Severity – Medium, Intention - Normal





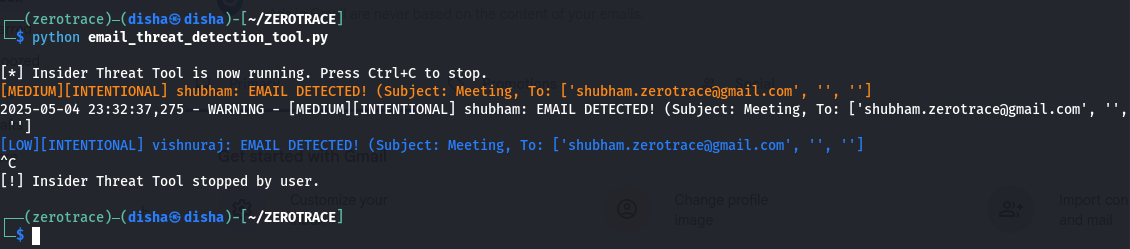
### Test case 10 :

An employee sends a base64 encoded string as a part of an email message to potentially hide certain information from detection tools. This flags for spoofing or obfuscation of emails.

Expected Outcome : Severity – Medium, Intention – Intentional

Actual Outcome : Severity – Medium, Intention - Intentional

## 

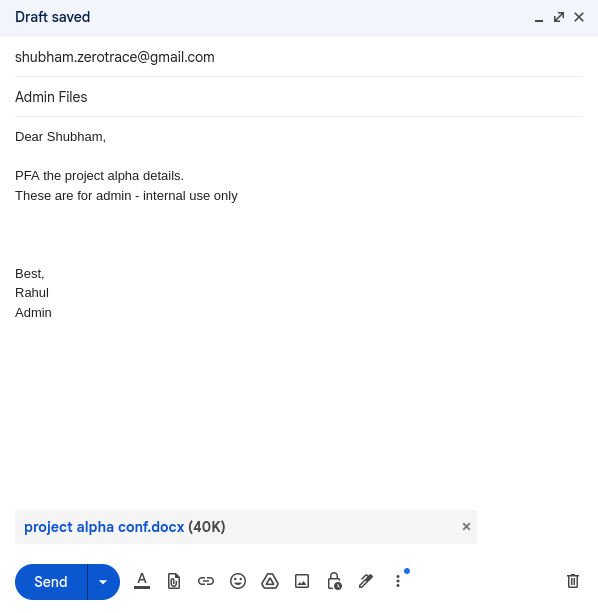


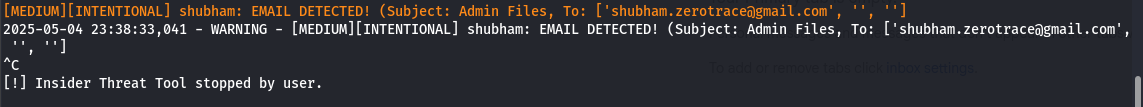
### Test case 11 :

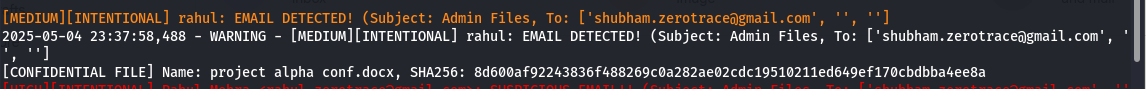
An employee (admin) sends an email to another employee (admin) containing a flagged and signed confidential attachment repeatedly where the recipient is unauthorized.

Expected Outcome : Severity – Medium, Intention – Intentional

Actual Outcome : Severity – Medium, Intention - Intentional







1. CONCLUSION

The current scope of the project includes insider threat detection for data exfiltration via email, particularly within small to medium-sized organizations. By combining static rule-based detection with fuzzy hashing, NLP-based content analysis, and Gmail API integration, the system provides real-time alerts, severity-based threat classification, and detailed logging. It effectively distinguishes between careless and malicious insider behavior while minimizing false positives. Experimental results demonstrate strong detection accuracy and responsiveness across both typical and edge-case scenarios. The tool offers a lightweight, customizable alternative to existing enterprise DLP solutions, making it especially valuable for resource-constrained environments.

1. FUTURE SCOPE

While the current system effectively detects insider threats via email, there are several areas where it can be expanded and enhanced:

* Machine Learning Integration: Incorporating supervised and unsupervised learning models can improve anomaly detection, especially for identifying subtle patterns of malicious behavior over time.
* Behavioral Baseline Profiling: Developing long-term user behavior profiles can enhance accuracy in distinguishing normal from suspicious activities.
* Advanced NLP Techniques: Leveraging transformer-based models (e.g., BERT, GPT) can improve detection of obfuscated or coded language used for covert data exfiltration.
* Multi-Channel Monitoring: Extending detection to other communication channels such as Slack, Teams, or USB transfers can offer a holistic insider threat detection platform.
* SIEM Integration: Seamless integration with Security Information and Event Management (SIEM) tools would allow for centralized alerting, response, and correlation with other network events.
* GUI Dashboard: A graphical interface can improve usability, enabling analysts to review alerts, logs, and statistics in real time.
* Automated Response Mechanisms: Adding automated mitigation actions like email blocking or user account flagging could strengthen defense in depth.

1. REFERENCES

# Bibliography

Bertino, E.; Ghinita, G. (2011) Towards mechanisms for detection and prevention of data exfiltration by insiders: keynote talk paper, Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, ACM, pp. 10–19. DOI: 10.1145/1966913.1966916

Schlicher, B.G.; MacIntyre, L.P.; Abercrombie, R.K. (2016) Towards reducing the data exfiltration for insider threats, 2016 49th Hawaii International Conference on System Sciences (HICSS), IEEE, pp. 2701–2710. DOI: 10.1109/HICSS.2016.345

Sabir, B.; Ullah, F.; Babar, M.A.; Gaire, R. (2021) Machine learning for detecting data exfiltration: A review, ACM Computing Surveys, 54 (7), Article 146, pp. 1–38. DOI: 10.1145/3442181

Hanley, M.; Montelibano, J. (2011) Insider Threat Control: Using Centralized Logging for Insider Threat Termination, Technical Report CMU/SEI-2011-TN-024, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA.

Essilfie-Conduah, N. (2019) A systems analysis of insider data exfiltration: A decentralized framework for disincentivizing and auditing data exfiltration, Master's Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.

Chung, M.H.; Yang, Y.; Wang, L.; Cento, G.; Jerath, K.; Raman, A.; Lie, D.; Chignell, M.H. (2023) Implementing Data Exfiltration Defense in Situ: A Survey of Countermeasures and Human Involvement, ACM Transactions on Privacy and Security, 26 (2),

PROGESS REPORT

PLAGIARISM REPORT

