Flipkart GRiD 5.0

Problem Statement Title: Compliance Monitoring and Enforcement through Log Analysis using Large Language Models (Infosec Engineering)

Team Name: Tech Fusion

Team members details

Team Name	Tech Fusion		
Institute Name/Names	Vellore Institute Of Technology, Vellore		
Team Members >	**1 (Leader)**	**2**	**3**
Name	Ketan Agrawal	Rishika Agrawal	Parimal Kothari
Batch	2024	2024	2024

Deliverables/Expectations for Level 2 (Idea + Code Submission)

Approach 1:With locally imported open source LLM model

Here's an overview of the approach and key components which we have used:

1.Document Loading:

- The code first defines a set of document loaders for various file types (e.g., CSV, PDF, HTML, TXT). These loaders are responsible for extracting text content from different document formats.

2.Loading and Splitting Documents:

- The **load_documents** function is used to load documents from a specified source directory. It searches for files with supported extensions and uses multiprocessing to load documents concurrently.
- Once loaded, the code splits the documents into smaller text chunks using a text splitter. This step is crucial for breaking down large documents into manageable pieces.

3.Embedding Creation:

- The code creates embeddings for each text chunk. It utilizes the Hugging Face Transformers library to create embeddings based on a specified model.
- Embeddings are representations of text chunks in a numerical form that capture semantic information.

4.Chroma Vector Store:

- The code uses the Chroma vector store to index and store these embeddings efficiently. Chroma allows for similarity searches, making it useful for retrieving relevant text chunks given a query.

5.Updating Existing Vector Store:

- If an existing vector store is found at the specified location, the code appends new documents to it. This is useful for incremental updates to the document collection.

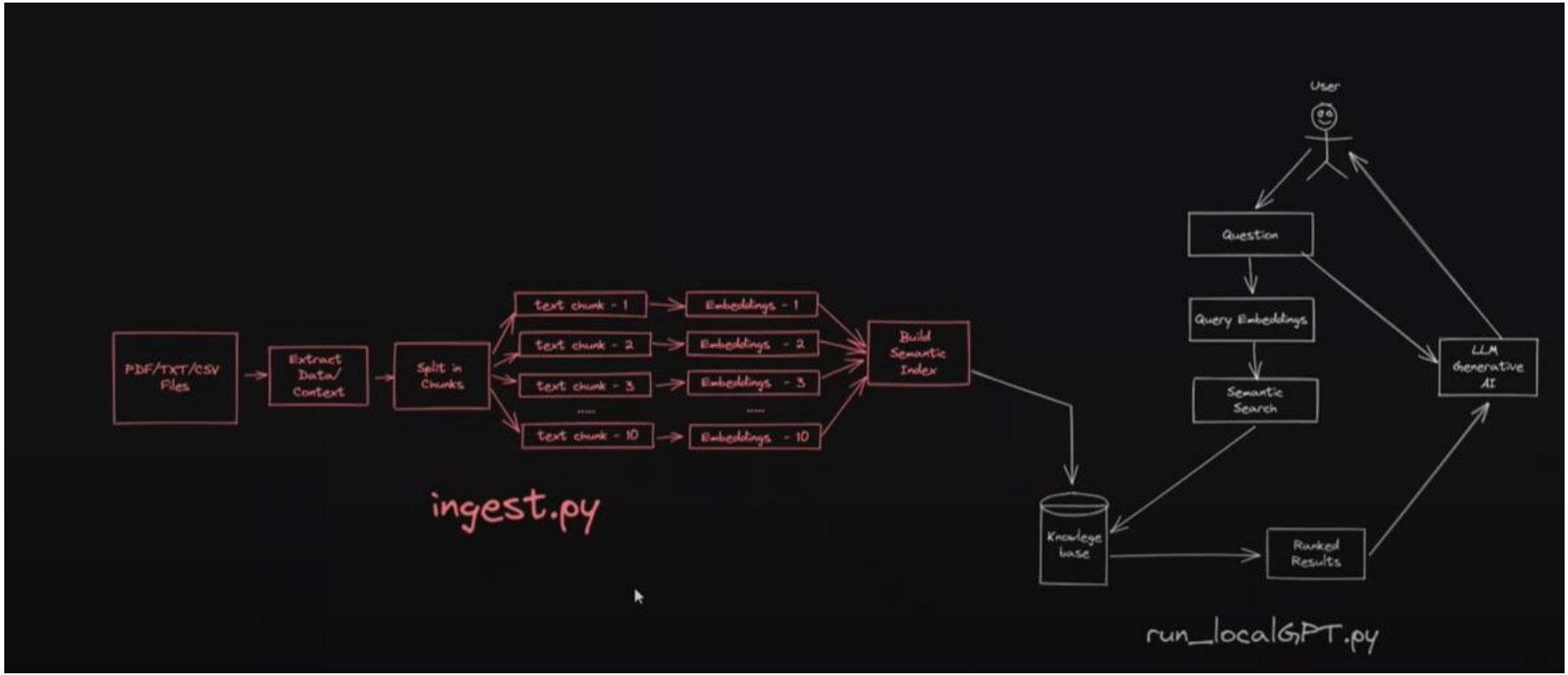
6.Persistence:

- After creating or updating the vector store, the code persists it to disk. This ensures that the vector store can be reused for future queries without re-embedding the entire document collection.

7.Main Function:

- The **main** function orchestrates the entire process. It first checks if a vector store already exists. If it does, it appends new documents; otherwise, it creates a new vector store.
- The documents are split, embedded, and added to the vector store.
- Once the process is complete, the vector store is persisted, and the code provides a message indicating that the ingestion is complete and that users can run queries for finding logs with anomalies in it.

Workflow


PDF/TXT/CSV Files
Extract Data/ Context
Split in Chunks
text chunk - 1
text chunk - 2
text chunk - 3
text chunk - 10
Embeddings - 1
Embeddings - 2
Embeddings - 3
Embeddings - 10
Build Semantic Index
ingest.py
User
Question
Query Embeddings
Semantic Search
LLM Generative AI
Knowlege base
Ranked Results
run_localGPT.py

Deliverables/Expectations for Level 2 (Idea + Code Submission)

Approach 2:With online AI apps LLM models like stackAI:

Here's an overview of the approach and key components which we have used:

1.Document uploading:

Upload the documents like rules.txt and opensshlogs.txt file in the document file upload node.

2.Providing input prompt:

“Analyze the opensshlogs file while keeping all the rules file content in mind and provide which logs are anomaly and give line number of the log with how to cure that anomaly or what actions to take to prevent that anomaly.” Provide this as input prompt for the input node.

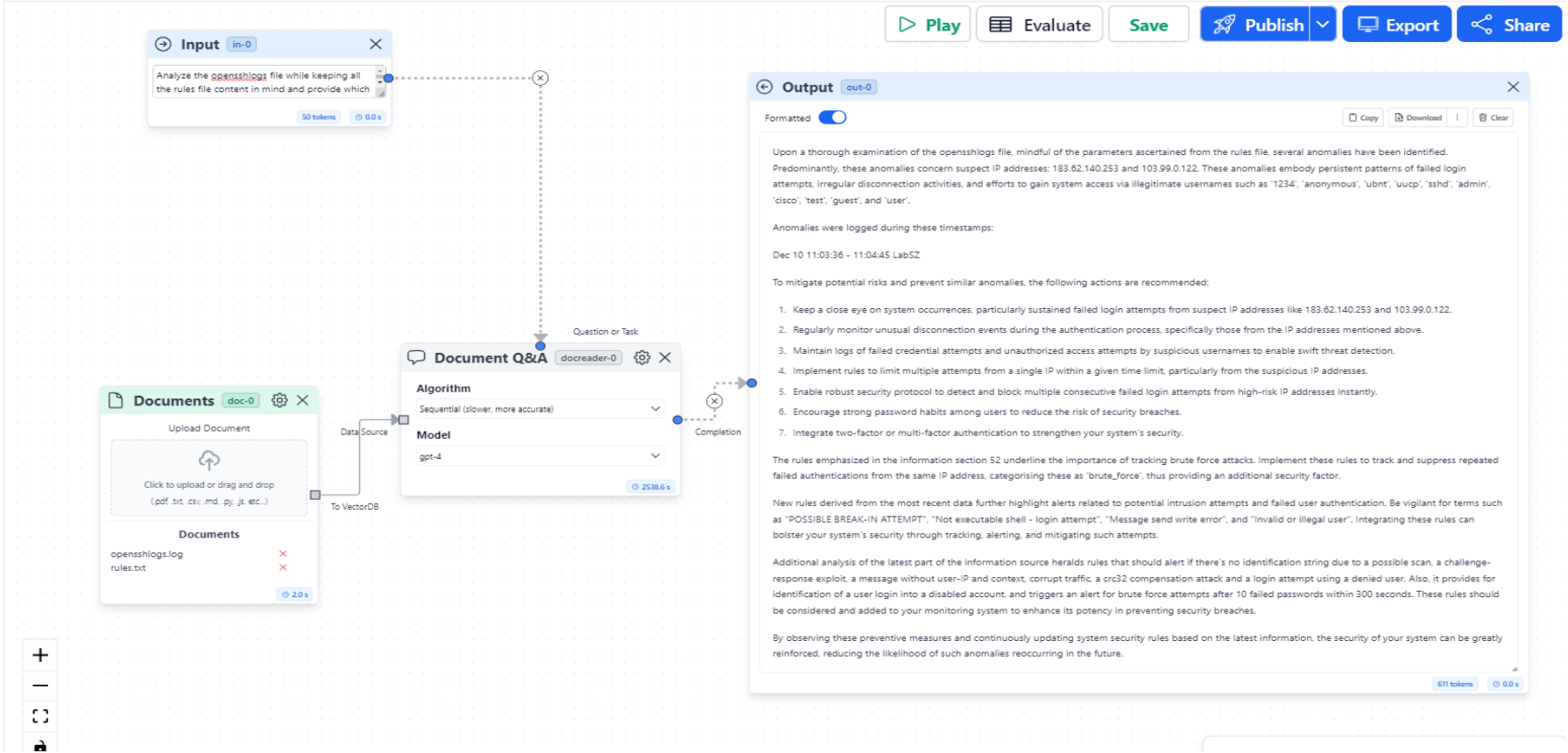
3.Document Q n A node:

Select the model to be gpt-4 and algorithm to be sequential as it is slower but more accurate.

4.Output:

Get the output as all the logs which are anomaly with remedies to be taken and what IP address to keep in check.

Workflow


Play
Evaluate
Save
Publish
Export
Share
Input
in-0
Analyze the opensshlogs file while keeping all the rules file content in mind and provide which
50 tokens
0.0 s
Question or Task
Document Q&A
docreader-0
Algorithm
Sequential (slower, more accurate)
Model
gpt-4
2538.6 s
Documents
doc-0
Upload Document
Click to upload or drag and drop
(.pdf, .txt, .csv, .md, .py, .js, etc..)
Documents
opensshlogs.log
rules.txt
2.0 s
Data Source
To VectorDB
Completion
Output
out-0
Formatted
Copy
Download
Clear
Upon a thorough examination of the opensshlogs file, mindful of the parameters ascertained from the rules file, several anomalies have been identified. Predominantly, these anomalies concern suspect IP addresses: 183.62.140.253 and 103.99.0.122. These anomalies embody persistent patterns of failed login attempts, irregular disconnection activities, and efforts to gain system access via illegitimate usernames such as '1234', 'anonymous', 'ubnt', 'uucp', 'sshd', 'admin', 'cisco', 'test', 'guest', and 'user'.
Anomalies were logged during these timestamps:
Dec 10 11:03:36 - 11:04:45 LabSZ
To mitigate potential risks and prevent similar anomalies, the following actions are recommended:
1. Keep a close eye on system occurrences, particularly sustained failed login attempts from suspect IP addresses like 183.62.140.253 and 103.99.0.122.
2. Regularly monitor unusual disconnection events during the authentication process, specifically those from the IP addresses mentioned above.
3. Maintain logs of failed credential attempts and unauthorized access attempts by suspicious usernames to enable swift threat detection.
4. Implement rules to limit multiple attempts from a single IP within a given time limit, particularly from the suspicious IP addresses.
5. Enable robust security protocol to detect and block multiple consecutive failed login attempts from high-risk IP addresses instantly.
6. Encourage strong password habits among users to reduce the risk of security breaches.
7. Integrate two-factor or multi-factor authentication to strengthen your system's security.
The rules emphasized in the information section 52 underline the importance of tracking brute force attacks. Implement these rules to track and suppress repeated failed authentications from the same IP address, categorising these as 'brute_force', thus providing an additional security factor.
New rules derived from the most recent data further highlight alerts related to potential intrusion attempts and failed user authentication. Be vigilant for terms such as "POSSIBLE BREAK-IN ATTEMPT", "Not executable shell - login attempt", "Message send write error", and "Invalid or illegal user". Integrating these rules can bolster your system's security through tracking, alerting, and mitigating such attempts.
Additional analysis of the latest part of the information source heralds rules that should alert if there's no identification string due to a possible scan, a challenge-response exploit, a message without user-IP and context, corrupt traffic, a crc32 compensation attack and a login attempt using a denied user. Also, it provides for identification of a user login into a disabled account, and triggers an alert for brute force attempts after 10 failed passwords within 300 seconds. These rules should be considered and added to your monitoring system to enhance its potency in preventing security breaches.
By observing these preventive measures and continuously updating system security rules based on the latest information, the security of your system can be greatly reinforced, reducing the likelihood of such anomalies reoccurring in the future.
611 tokens
0.0 s

Glossary

- Rules.txt- for rules to keep in check all the logs of openssh.
- Opensshlogs.txt- all the openssh logs collected with activities which are harmful also.
- Github repository link for code: https://github.com/Rishika631/Flipkart_Grid

Use-cases

P0 (High Impact):

1. **Rule Definition and Inference:** Develop a rule engine that can distill compliance rules and infer relationships between rules and parameters in the logs and policies. Ensure accurate rule interpretation.
2. **Flexible Log Format Handling:** Create a system that can handle various log formats (e.g., CSV, text, PDF) commonly used in different environments and systems. Ensure that log parsing is robust and adaptable.
3. **Scalability:** Design the system to handle large volumes of logs and text data efficiently. Ensure that it can scale horizontally to accommodate growing data volumes.

P1 (Medium Impact):

4. **Custom Rule Sets:** Allow users to define custom compliance rule sets based on their organization's specific security policies and standards. Provide a user-friendly interface for rule configuration.
5. **Actionable Insights:** Generate actionable insights and recommendations for remediation when compliance breaches are detected. Provide detailed information on how to fix issues and prevent future violations.

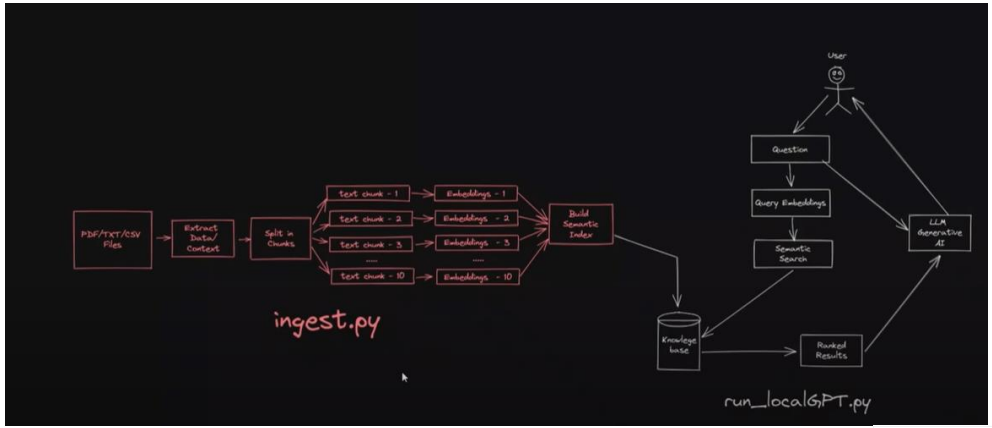
P2 (Low Impact):

6. **User Training and Documentation:** Provide training materials and documentation to educate users on how to effectively use the compliance monitoring system.

P3 (Enhancements):

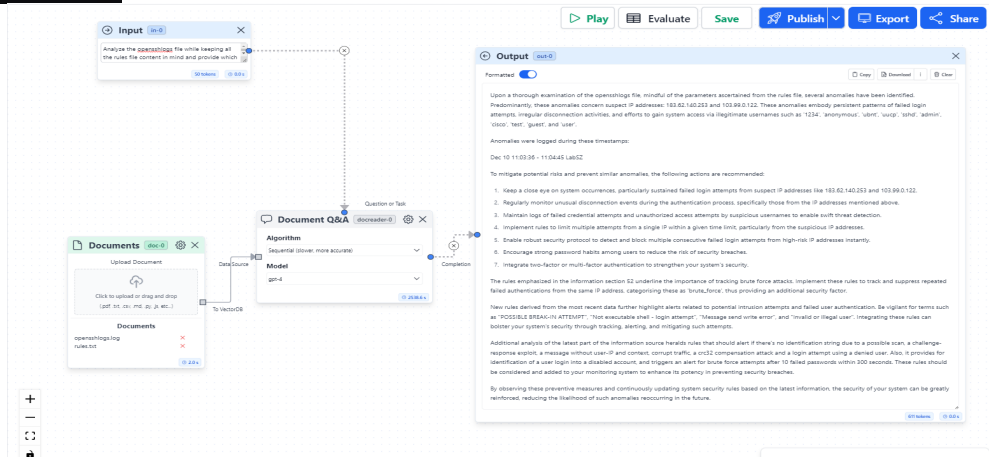
7. **Machine Learning for Anomaly Detection:** Explore the use of machine learning techniques, in addition to LLMs, for anomaly detection in logs and user behavior.
8. **Automated Remediation:** Investigate the possibility of implementing automated remediation actions for common compliance violations, with appropriate safeguards and approvals.

Solution statement/ Proposed approach



Approach 1

Approach 2



Limitations

1.Resource Intensive: LLMs, such as GPT-4 or its alternatives, can be resource-intensive in terms of computational power and memory. Running these models for log analysis on large datasets may require significant hardware resources.

2.Scalability: While scalability is mentioned as a requirement, it can still be a challenge. Scaling up LLM-based systems to handle massive volumes of logs in real-time can be complex and costly.

3.Data Privacy and Security: Handling sensitive log data and compliance information requires robust data privacy and security measures. Ensuring that the system complies with data protection regulations can be a significant challenge.

4. Cost: Implementing and maintaining a system with LLMs and associated technologies can be costly, both in terms of infrastructure and human resources.

5.AI Bias: LLMs can inherit biases present in their training data, which may impact the fairness of compliance checks. Mitigating bias is an ongoing concern in AI.

6.Maintenance Overhead: Ongoing maintenance and updates to keep the system effective and up-to-date can be a significant overhead.

Future Scope

1. **Enhanced Reporting and Dashboards:** Create more comprehensive and customizable reporting and dashboard features that provide stakeholders with real-time insights into compliance status and trends.

2. **User Behavior Analysis:** Incorporate user behavior analytics to detect anomalous user activities and potential insider threats. This includes profiling user behavior to identify deviations from normal patterns.

3. **Threat Intelligence Integration:** Integrate threat intelligence feeds and databases to enhance the system's ability to detect compliance breaches related to known threats and vulnerabilities.

4. **Regulatory Compliance Automation:** Develop features that automate the process of generating compliance reports and documentation required for regulatory audits and assessments.

5. **Integration with ITSM and GRC Tools:** Integrate with IT Service Management (ITSM) and Governance, Risk, and Compliance (GRC) tools to streamline incident management, risk assessment, and compliance tracking processes.

6. **Continuous Feedback Loop:** Strengthen the feedback loop for user feedback, allowing the system to continuously learn and improve its compliance rule interpretation and analysis capabilities.

Flipkart

GRiD

5.0

Thank You