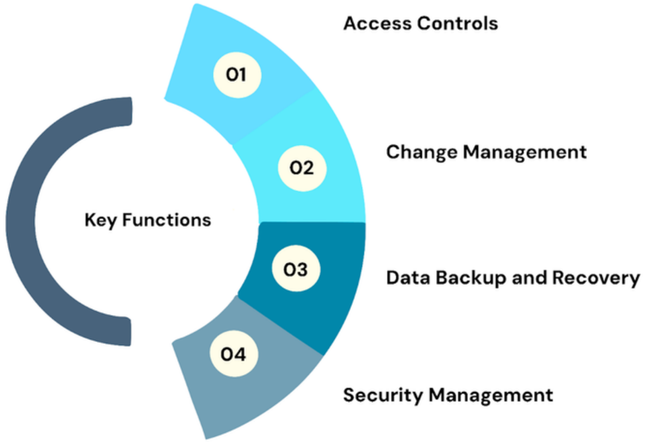
ITGC ACCESS CONTROL AUDIT USING PYTHON & POWER BI

PRESENTED BY SOUBARNO BHATTACHARYA

Project Overview

Power BI



Component	Details
Objective	Identify and visualize access violations including:• Post-Termination Access• After-Hours Logins
Scope of Audit	Merge system access logs with HR master data to detect unauthorized or risky access patterns.
Tools Used	• Python (Pandas) for data cleaning & merging• Power BI for interactive dashboards• Excel (employee master simulation)
Why ITGC Matters	IT General Controls (ITGC) are critical for SOX 404 compliance and internal audit assurance, especially in preventing unauthorized system access

Data Sources & Merging

1. Access Logs

Source: Public dataset from Kaggle

What it includes:

employee_id, timestamp, access_point, access_type

Purpose:

Track when and where each employee accessed the system

Detect unusual login patterns

2. Employee Master Data

Source: Manually created in Excel (simulates HR system)

What it includes:

employee_id, Name, Department, Role, Join_Date, Termination_Date, status

Purpose:

Provide employment context

Identify if employee was active, terminated, or in a sensitive role

3. Merged Dataset

Created By: Merging Access Logs + Employee Master in Python

Why It's Critical:

Combine system activity with employee status

Enables detection of:

Post-Termination Access

After-Hours Logins

Violations by Department or Role

Reflects real-world audit workflows used in ITGC & SOX audits

Data Cleaning & Preparation in Python

Data Cleaning & Formatting

Goal: Ensure date/time consistency and clean column structure

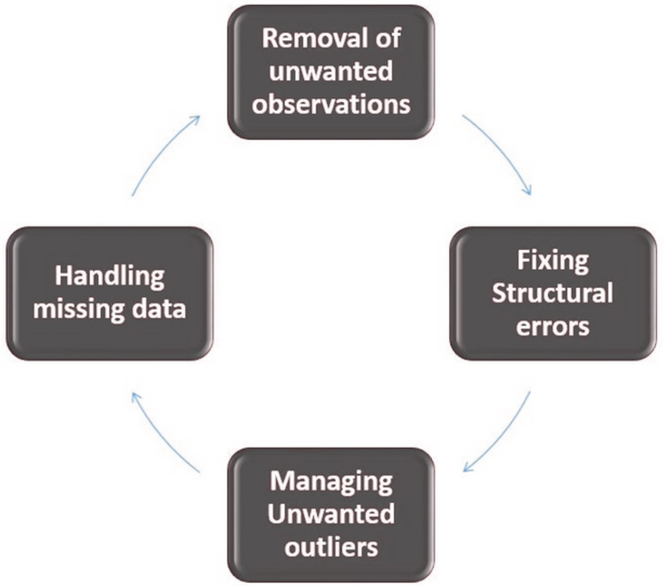
Parsed timestamp column to datetime

Extracted hour, day_of_week, and date

Removed nulls and duplicates

Standardized employee IDs and column names

Saved cleaned logs as cleaned_logs.csv



Final Merged Dataset

Cleaned access logs were merged with employee master data using employee_id. This added role, department, and employment status to each login. The final dataset enabled accurate detection of access control violations.

Feature Engineering (Audit Flags)

Created after_hours_access (True if login before 7 AM or after 8 PM) and post_termination_access (True if login occurred after termination date) using Python logic.

These flags are essential audit markers used to detect ITGC violations.

Power BI Dashboard – Violation Insights

Post_Termination_Access Violations

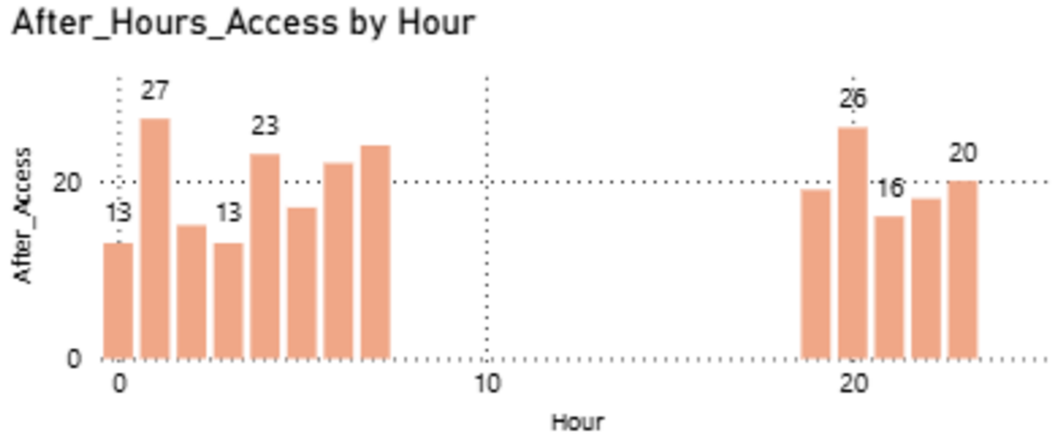
166

Post-Termination Access:
Displays the total number of login attempts made after employee termination.

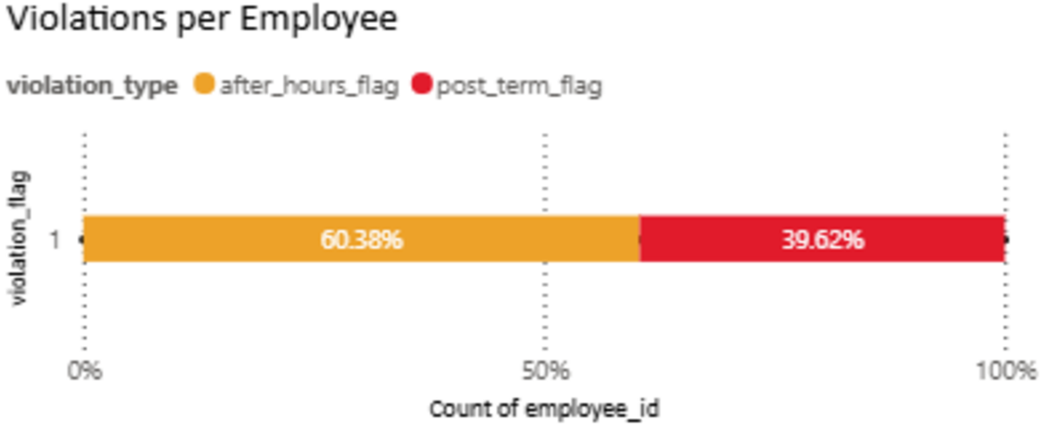
After-Hours Access Violations

253

After-Hours Access:
Shows how many logins occurred outside standard working hours (before 7 AM or after 8 PM).



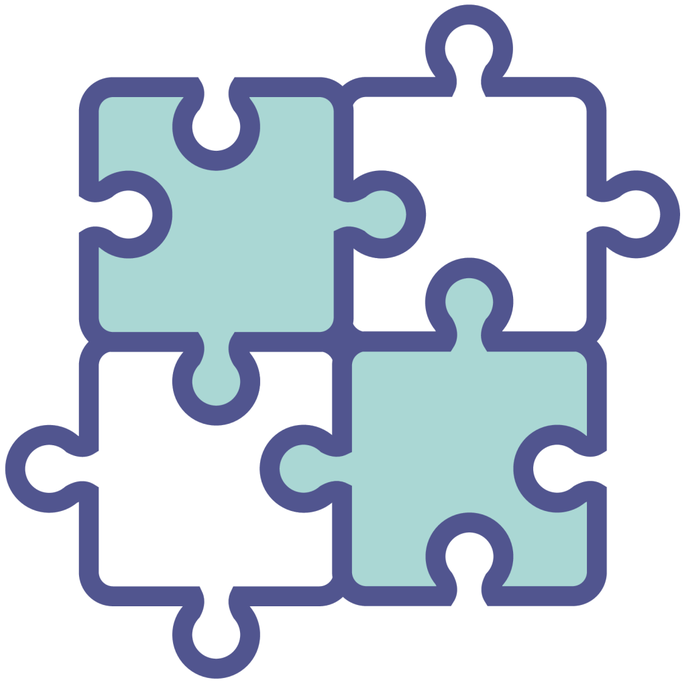
The column chart highlights login activity during non-business hours.
Peaks at 1 AM, 3 AM, and 9 PM indicate multiple access events occurring at unusual times, suggesting elevated audit risk.



The stacked bar chart shows the proportion of ITGC violations across employees.
🔸 60.38% were due to after-hours access, while
🔴 39.62% resulted from post-termination logins.
This breakdown highlights that most violations stem from employees accessing systems outside approved working hours.

S.W.O.T

swot analysis



S

Strengths

Simulates real-world ITGC audit using Python & Power BI

Violations clearly visualized with audit-ready KPIs

W

Weaknesses

Employee master data manually created

RBAC not implemented due to scope

T

Threats

Assumptions (e.g., working hours) may not align with all orgs

Public dataset may not reflect actual enterprise risks

O

Opportunities

Extend to include privilege access & RBAC audits

Can be scaled for continuous monitoring in real audits

THANK YOU

VISIT ME

+91 7439138833

sou2608barno@gmail.com