

2FA Registration Process Automation & Security Enforcement

submitted by

Arijit Maity

Class Roll No: 002210503027

Exam Roll: MCA244026

Master of Computer Application

under the supervision of

Prof. Debotosh Bhattacharjee

in the

Department of Computer Science and Engineering

Faculty of Engineering and Technology

Jadavpur University

2024

Faculty of Engineering and Technology
Jadavpur University

CERTIFICATE OF RECOMMENDATION

To Whom It May Concern

I hereby recommend that the project report entitled “2FA Registration Process Automation & Security Enforcement” has been carried out by Arijit Maity (Reg. No.: 163619 of 2022-23, Roll No.: 002210503027, Exam Roll: MCA244026), under my guidance and supervision and be accepted in partial fulfilment of the requirement for the degree of Master of COMPUTER APPLICATION in DEPARTMENT of COMPUTER SCIENCE and ENGINEERING, JADAVPUR UNIVERSITY during the academic year 2023-2024.

Prof. Debotosh Bhattacharjee
Project Supervisor
Dept. of Computer Science and
Engineering
Jadavpur University

Prof. (Dr.) Nandini Mukhopadhyay
Head of the Department
Dept. of Computer Science & Engineering
Jadavpur University

Prof. Dipak Laha
Dean, Faculty Council
of Engineering & Technology
Jadavpur University, Kolkata-032

Faculty of Engineering and Technology
Jadavpur University

CERTIFICATE OF APPROVAL

This is to certify that the project report entitled “**2FA Registration Process Automation & Security Enforcement**” is a bonafide record of work carried out by ARIJIT MAITY, in fulfillment of the requirements for the award of the degree of Master of Computer Application in the Department of Computer Science and Engineering, Jadavpur University. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the project report only for the purpose for which it has been submitted.

Signature of Examiner 1

Date :

Signature of Examiner 2

Date :

Faculty of Engineering and Technology
Jadavpur University

**DECLARATION OF COMPLIANCE WITH
ACADEMIC ETHICS**

I hereby declare that this project report entitled "2FA Registration Process Automation & Security Enforcement" is an outcome of my genuine and independent work, as part of my Master of Computer Application program.

All information in this document has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, all materials and results that are not original to this work have been fully cited and referenced.

Arijit Maity

Registration No.: 163619 of 2022-23

Examination Roll No.: MCA244026

Thesis Title :- 2FA Registration Process Automation & Security Enforcement

Signature with Date

ACKNOWLEDGEMENT

I wish to record my heartfelt gratitude towards all the people who have helped me in the completion of this project.

I offer sincere thanks to my internal guide Prof. Debotosh Bhattacharjee and my external guide, Mr. Tarun Joseph of TI, for their kind support, inspiration, and thoughtful insights.

I sincerely thank Professor Nandini Mukhopadhyay, Head of the Department of Computer Science and Engineering at Jadavpur University, for her continuous support and guidance throughout the project.

I am extremely thankful to Texas Instruments and the Department of Computer Science and Engineering at Jadavpur University for providing all the facilities and necessary help towards the development of this project.

Arijit Maity

Class Roll No.: 002210503027

Examination Roll No.: MCA244026

Department of Computer Science & Engineering

Jadavpur University

CONTENT

<u>No</u>	<u>Topic</u>	<u>Page No.</u>
1	Introduction.....	1-3
1.1	Background of 2FA Authentication.....	1
1.2	Definition and Purpose.....	1
1.3	Evolution of 2FA	1
1.4	Importance of 2FA in Security.....	2
1.5	Current Challenges in 2FA Implementation.....	2
1.6	User Convenience Vs Security.....	2
1.7	Vulnerability in Traditional 2FA Methods.....	2
1.8	Motivation of 2FA Registration Process Automation.....	3
2	Introduction to Entrust.....	3-5
2.1	What is Entrust.....	3
2.2	Services Offered by Entrust.....	3
2.3	Importance of Entrust in Digital Security.....	4
2.4	Future Trends in Digital Security and Entrust Role.....	4
2.5	Entrust Architecture.....	5
3	Problem Statement.....	6-8
3.1	Current Manual Process of 2FA Registration.....	6
3.2	Challenges and Limitation of Manual Process.....	6
3.3	Need of Automation of 2FA Registration.....	7
3.4	Important of Extracting Key Metrics in 2FA.....	7
3.5	Expected Outcome.....	8
4	Design of Solutions.....	9-27
4.1	Automation of 2FA Registration.....	9-15
4.1.1	Overview.....	9
4.1.2	Adding Users in AD Group.....	10-13
4.1.3	Removing Users in AD Group	13-15

4.2	Self-Service Portal.....	16-22
4.2.1	Need Of Self-Service Portal.....	16
4.2.2	Workflow of Self-Service Portal.....	16-18
4.2.3	Flask Api Endpoints.....	18-19
4.2.4	Types of View	20-22
4.3	Extracting Metrics.....	23-27
4.3.1	Need of Metrics.....	23
4.3.2	Workflow.....	24-25
4.3.3	Endpoints.....	26
4.3.4	Database Design.....	27
5	Deployments	28
6	Result and Analysis.....	29
7	Conclusion and Scope of future work.....	30
8	References.....	32

ABSTRACT

This thesis presents the development and implementation of a system for automating the Two-Factor Authentication (2FA) process registration and extracting key metrics. The project aims to streamline and enhance the security of user authentication by automating the registration process for 2FA and extracting meaningful metrics for analysis. The system utilizes cutting-edge technologies and methodologies to achieve these objectives, including but not limited to, machine learning algorithms for data analysis, secure communication protocols for authentication, and user-friendly interfaces for seamless user interaction. The results demonstrate the feasibility and effectiveness of the proposed system in improving the security and efficiency of the 2FA process

Chapter 1. Introduction

1.1 Background of Two-Factor Authentication (2FA):

In today's digital landscape, where cyber threats are prevalent, ensuring the security of online accounts and sensitive information is paramount. Two-Factor Authentication (2FA) has emerged as a crucial security measure to protect against unauthorized access. Unlike traditional single-factor authentication methods that rely solely on a password, 2FA adds an additional layer of security by requiring users to provide a second form of identification. This second factor is typically something the user possesses, such as a smartphone or a hardware token.

1.2 Definition and Purpose:

Two-Factor Authentication (2FA) is a security process in which users provide two different authentication factors to verify themselves. These factors may include something they know (e.g., a password), something they have (e.g., a smartphone or token), or something they are (e.g., biometric data like fingerprints or facial recognition). The purpose of 2FA is to add an extra layer of security to user accounts, making them less vulnerable to unauthorized access, even if the password is compromised.

1.3 Evolution of 2FA:

The concept of 2FA has evolved over time in response to the growing need for stronger authentication methods. It has progressed from simple token-based systems to more sophisticated methods, such as biometric authentication and push notifications to mobile devices. These advancements have significantly enhanced the security of 2FA and made it more user-friendly.

1.4 Importance of 2FA in Security:

The importance of 2FA in enhancing security cannot be overstated. With the increasing frequency of data breaches and cyber-attacks, traditional password-based authentication methods are no longer sufficient to protect sensitive information. 2FA provides an additional layer of security that significantly reduces the risk of unauthorized access, making it an essential component of modern security practices.

1.5 Current Challenges in 2FA Implementation:

While 2FA is an effective security measure, its implementation comes with challenges. One of the primary challenges is finding the right balance between security and user convenience. Complex 2FA methods may be more secure but cumbersome for users, leading to lower adoption rates. Conversely, overly simplistic methods may not provide adequate security.

1.6 User Convenience vs Security:

Achieving a balance between user convenience and security is crucial for the successful implementation of 2FA. Users are more likely to adopt 2FA if it is easy to use and does not significantly disrupt their workflow. This highlights the importance of designing 2FA systems that are secure and user-friendly.

1.7 Vulnerabilities in Traditional 2FA Methods:

Traditional 2FA methods, such as SMS-based authentication and token-based systems, are not without vulnerabilities. SMS-based authentication, for example, is susceptible to SIM swapping attacks, where an attacker fraudulently obtains a user's SIM card to intercept SMS codes. Similarly, token-based systems can be vulnerable if the token is lost or stolen.

1.8 Motivation for 2FA Registration Process Automation:

The motivation behind automating the 2FA registration process is to improve the user experience and make 2FA more accessible to a wider audience. By automating the registration process, users can easily set up 2FA without the need for manual intervention, thus reducing the barrier to adoption. This can lead to higher adoption rates and improved security overall.

Chapter 2. Introduction to Entrust

2.1 What is Entrust?

Entrust is a globally recognized leader in providing digital security solutions that help organizations protect their digital assets. With a rich history spanning several decades, Entrust has established itself as a trusted partner for organizations seeking to enhance their security posture in an increasingly digital world. Entrust offers a wide range of solutions, including identity and access management, secure communications, and fraud detection.

2.2 Services Offered by Entrust

Entrust offers a comprehensive suite of services designed to address the evolving challenges of digital security. Some of the key services offered by Entrust include:

- ❖ **Certificate Management:** Entrust provides solutions for managing digital certificates, including issuance, renewal, and revocation.
- ❖ **Authentication Solutions:** Entrust offers a range of authentication solutions, including Two-Factor Authentication (2FA) and biometric authentication.
- ❖ **Secure Email Services:** Entrust provides secure email solutions that encrypt emails to protect sensitive information.
- ❖ **Secure Digital Signatures:** Entrust offers solutions for creating and verifying digital signatures, ensuring the integrity and authenticity of digital documents.

2.3 Importance of Entrust in Digital Security:

Entrust plays a crucial role in enhancing digital security for organizations by providing robust solutions that help protect against cyber threats. By leveraging Entrust's services, organizations can enhance their security posture, protect their digital assets, and comply with regulatory requirements related to security and data protection.

2.4 Future Trends in Digital Security & Entrust Role:

The field of digital security is constantly evolving, with new threats and challenges emerging regularly. Entrust is at the forefront of these advancements, continuously innovating to provide cutting-edge solutions that address the latest threats. Entrust helps organizations navigate the complex digital security landscape and protect their digital assets by staying ahead of the curve.

2.5 Entrust Architecture:

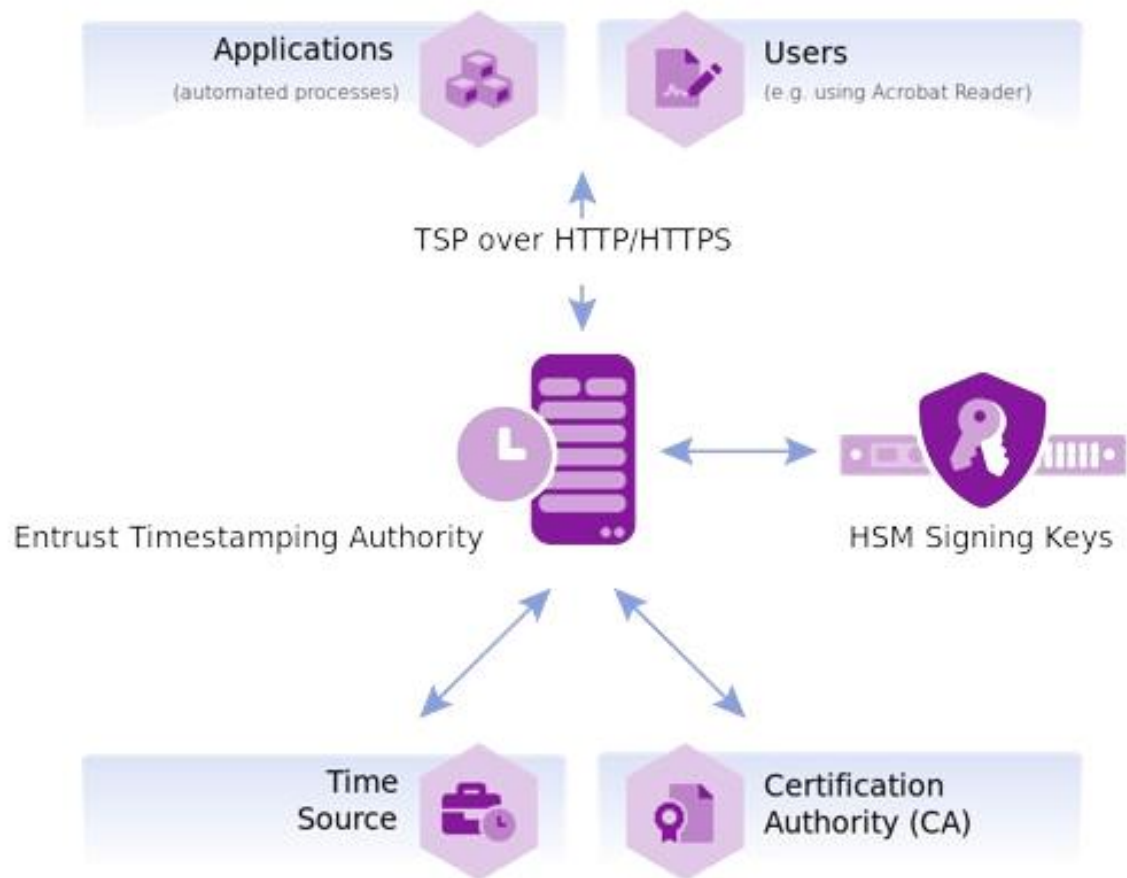
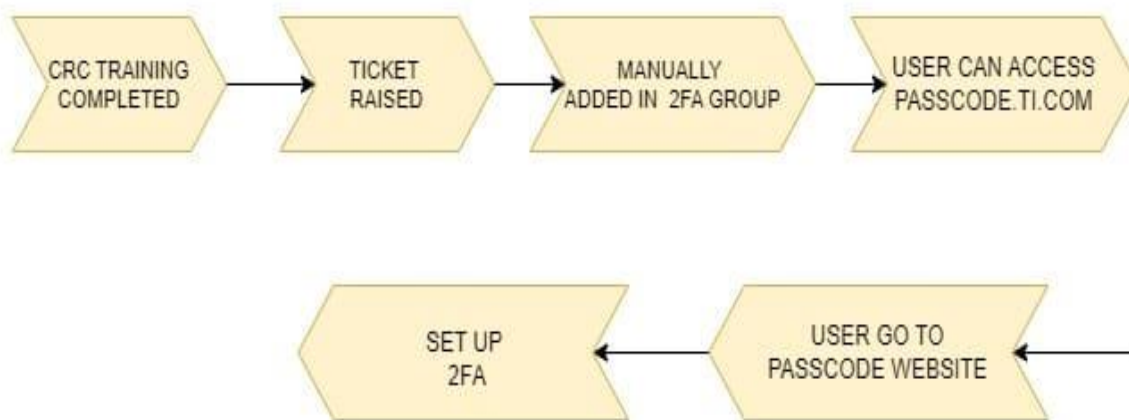


Diagram of Entrust Architecture

Chapter 3. Problem Statement

3.1 Current Manual Process of 2FA Registration:

Currently, the process of registering for Two-Factor Authentication (2FA) in our organization is manual and cumbersome. To enable 2FA, users are required to undergo training, raise a ticket, and then be manually added to a 2FA Active Directory group. This process is time-consuming and prone to errors, leading to delays in enabling 2FA for users.



Flowchart Of Current Manual Process

3.2 Challenges and Limitations of the Manual Process:

The manual process of 2FA registration poses several challenges and limitations, including:

- ❖ **Time-Consuming:** The manual process requires multiple steps and manual intervention, leading to delays in enabling 2FA for users.
- ❖ **Error-Prone:** Manual processes are prone to errors, such as data entry mistakes or delays in processing requests.

- ❖ **Lack of Scalability:** The manual process becomes inefficient as the number of users requiring 2FA increases, leading to delays and bottlenecks in the process.

3.3 Need for Automation of 2FA Registration:

There is a pressing need to automate the 2FA registration process to overcome the challenges and limitations of the current manual process. Automation will not only streamline the process but also improve efficiency, reduce errors, and enhance security by enabling 2FA for users in a timely manner.

3.4 Importance of Extracting Key Metrics in 2FA:

In addition to automating the 2FA registration process, it is also essential to extract key metrics to monitor and analyze the effectiveness of the 2FA implementation. Currently, there is a lack of metrics to track important aspects such as the number of user accounts locked due to incorrect authentication attempts, active users using 2FA, and incomplete registrations. Extracting these metrics will provide valuable insights into the security and usability of the 2FA system, enabling us to make informed decisions and improve the overall user experience.

3.5 Expected Outcomes:

- ❖ **Automation of 2FA Registration Process:**

The primary expected outcome of this project is the successful automation of the 2FA registration process. By implementing an automated system, users will be able to enable 2FA without the need for manual intervention. This will streamline the registration process, reduce delays, and improve overall efficiency.

- ❖ **Improved Efficiency and User Experience:** Automating the 2FA registration process will lead to improved efficiency and a better user experience. Users will no longer have to undergo training, raise tickets, or wait for manual approval to enable 2FA. Instead, they will be able to enable 2FA quickly and easily, enhancing the overall usability of the system.

- ❖ **Enhanced Security:** Automating the 2FA registration process will also enhance security by reducing the risk of human error and ensuring that 2FA is enabled consistently and correctly for all users. This will help protect sensitive information and prevent unauthorized access to accounts, enhancing the organization's overall security posture.

- ❖ **Extraction and Analysis of Key Metrics:** Another expected outcome of this project is extracting and analyzing key metrics related to 2FA usage. By extracting metrics such as the number of user accounts locked due to incorrect authentication attempts, the number of active users using 2FA, and the number of incomplete registrations, we will be able to gain valuable insights into the effectiveness of the 2FA implementation. This will enable us to make informed decisions and further improve the 2FA system.

Chapter 4. DESIGN OF SOLUTIONS

4.1 Automation Of 2FA Registration:

4.1.1 Overview:

The automation of the 2FA registration process involves two main tasks: adding users who have completed training in the last 24 hours to an Active Directory (AD) group for 2FA setup, and removing users who have been in the AD group for more than 7 days.

- Adding Users to the AD Group:

To automate the addition of users to the AD group, we will implement a script that queries the training database to retrieve the employee IDs of users who have completed training in the last 24 hours. This script will then add these users to the designated AD group for 2FA setup.

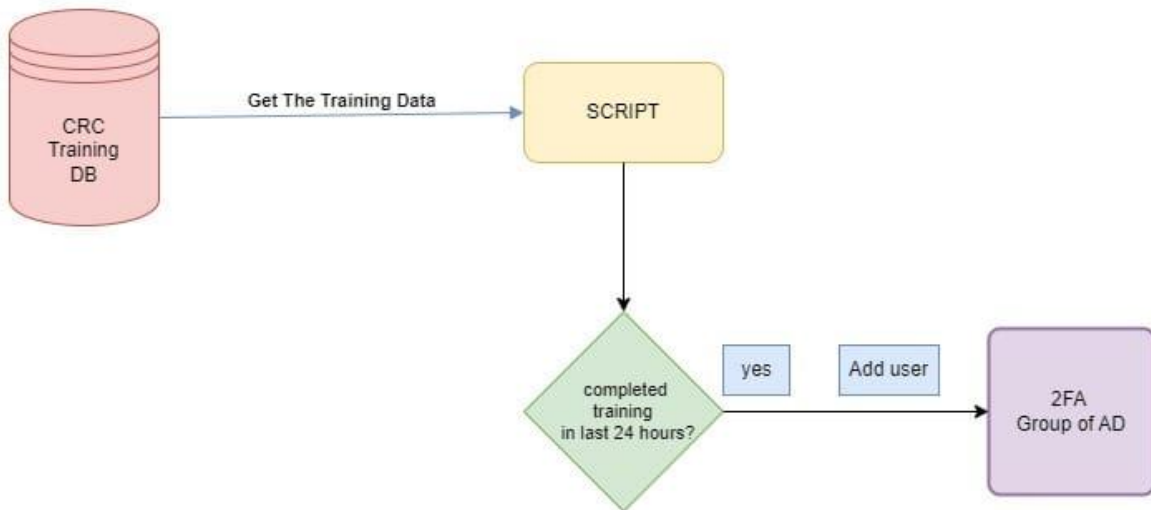
- Removing Users from the AD Group:

To automate the removal of users from the AD group, we will implement a script that queries PowerShell to retrieve the date when each user was added to the AD group. This script will then compare the current date with the date when each user was added and remove users who have been in the AD group for more than seven days.

4.1.2 Adding Users to the AD Group:

- The script will query the training database to retrieve employee IDs of users who have completed training in the last 24 hours.

- The script will add the user to the designated AD group for 2FA setup using PowerShell commands for each retrieved employee ID.
- The script will run on a scheduled basis (e.g., daily) to ensure that new users are promptly added to the AD group after completing training.



Flowchart Of Adding Into 2FA Group

Adding script Code:

```

import os
import logging
import datetime
from ldap3.extend.microsoft.addMembersToGroups import ad_add_members_to_groups
from dotenv import load_dotenv
from establish_the_connection import establish_the_connection_with_server_pre_prod
from get_distinguished_names_list_from_employee_ids import get_distinguished_names_list_from_employee_ids
from pull_distinguished_names_of_group import pull_distinguished_names_of_group
from get_training_completed_employee_ids_list import get_training_completed_employee_ids_list
current_date_time=datetime.datetime.now().strftime("%d_%m_%Y_%H_%M_%S")
#log file name
file_name_for_log=f"./adding_in_the_group{current_date_time}.log"

```

```

#log file configuration
#setting to debug so that we can log info,error,warning message also
logging.basicConfig(filename=file_name_for_log,level=logging.DEBUG,format='%(asctime)s -%(levelname)s -%(message)s')
load_dotenv()
oracledb.init_oracle_client()

host_name=os.getenv("ORACLE_HOST_NAME")
port_value=os.getenv("PORT_VALUE")
service_name=os.getenv("ORACLE_SERVICE_NAME")
username=os.getenv("ORACLE_USER_NAME")
password=os.getenv("ORACLE_PASSWORD")
service_account_name=os.getenv("LDAP_SERVICE_ACCOUNT_NAME_PRE")
service_account_password=os.getenv("LDAP_SERVICE_ACCOUNT_PASSWORD_PRE")
server_name=os.getenv("LDAP_SERVER_NAME_PRE")

SEARCH_BASE_FOR_DISTINGUISHED_NAME=os.getenv("SEARCH_BASE_FOR_DISTINGUISHED_NAME")
)
GROUP_NAME_FOR_ADDING=os.getenv("GROUP_NAME_FOR_ADDING")

def establish_the_connection_with_server_pre_prod():
    server = Server(server_name, get_info=ALL)
    #establishing the connection with pre prod server
    conn = Connection(server, user=service_account_name,
password=service_account_password, auto_bind=True)
    conn.bind()
    return conn
#get distinguished names list of employee id list
def
get_distinguished_names_list_from_employee_ids(conn,employee_ids_list,search_base
_for_distinguished_name):
    employee_distinguished_names_list=[]
    #iterate over the list and extract the distinguished name
    for employee_id in employee_ids_list:
        conn.search(search_base={search_base_for_distinguished_name},
                    search_filter=f"(SamAccountName={employee_id})",
                    search_scope=SUBTREE,
                    attributes=[ALL_ATTRIBUTES],
                    size_limit=0)
        #this will handle the situation where we get employee ids that actually
does not exist at all
        #in AD server
        if(conn.entries):

```

```

        employee_distinguished_names_list.append(conn.entries[0]['distinguish
edName'].value)
    return employee_distinguished_names_list

def get_training_completed_employee_ids_list():
    group_id_find_query="""select employee_id from employee_details order by
dbms_random.value FETCH NEXT 5 ROWS ONLY"""
    connection,cursor=establish_connection_to_db()
    result=cursor.execute(group_id_find_query).fetchall()
    employee_ids_list=[]
    for data in result:
        employee_ids_list.append(data[0])
    close_connection_from_db(connection,cursor)
    return employee_ids_list

def pull_distinguished_names_of_group(conn,group_name):
    conn.search(search_base=f'{group_name}',
                search_filter='(objectClass=*)',
                search_scope=SUBTREE,
                attributes=[ALL_ATTRIBUTES],
                size_limit=0)
    if 'member' in conn.entries[0]:
        distinguished_names_list=conn.entries[0].member
    else:distinguished_names_list=[]
    return distinguished_names_list

def main():
    logging.info("The script for adding in the group started")
    #establish the connection
    conn=establish_the_connection_with_server_pre_prod()
    #employee ids list for adding
    employee_ids_list=get_training_completed_employee_ids_list()

    #distinguished names list
    #if there is one or more than employees
    #then only we will extract the distinguished name and add in the group
    if(len(employee_ids_list)):
        distinguished_names_list_of_employee_ids=get_distinguished_names_list_from_
employee_ids(conn,employee_ids_list,SEARCH_BASE_FOR_DISTINGUISHED_NAME)
        for employee_dn in distinguished_names_list_of_employee_ids:
            print(employee_dn)
    # print(len(distinguished_names_list_of_employee_ids))

```

```

# #adding members to the group
if(len(distinguished_names_list_of_employee_ids)):
    ad_add_members_to_groups(conn,distinguished_names_list_of_employee_ids,GR
OUP_NAME_FOR_ADDING)

    distinguished_name_of_the_group=pull_distinguished_names_of_group(conn,GROUP_
NAME_FOR_ADDING)
    print(len(distinguished_name_of_the_group))

#close the connection
conn.unbind()
logging.info("The script for adding in the group ended")
logging.info("Total time taken:")

if __name__=="__main__":
    main()

```

4.1.3 Removing Users from the AD Group:

We will use a two-step process to automate removing users from the Active Directory (AD) group for 2FA setup. First, we will write a script in PowerShell to extract the adding date of users from the AD group and insert this data into a 2FA database. Second, we will write a script in Unix to extract the data from the 2FA database and remove users from the AD group who have been in the group for more than seven days.

Step1:

Extracting and Inserting Adding Date into 2FA Database (PowerShell Script):

The PowerShell script will query the AD group for 2FA setup and retrieve the adding date of each user in the group.

For each user, the script will insert the adding date into a 2FA database table along with the user's employee ID.

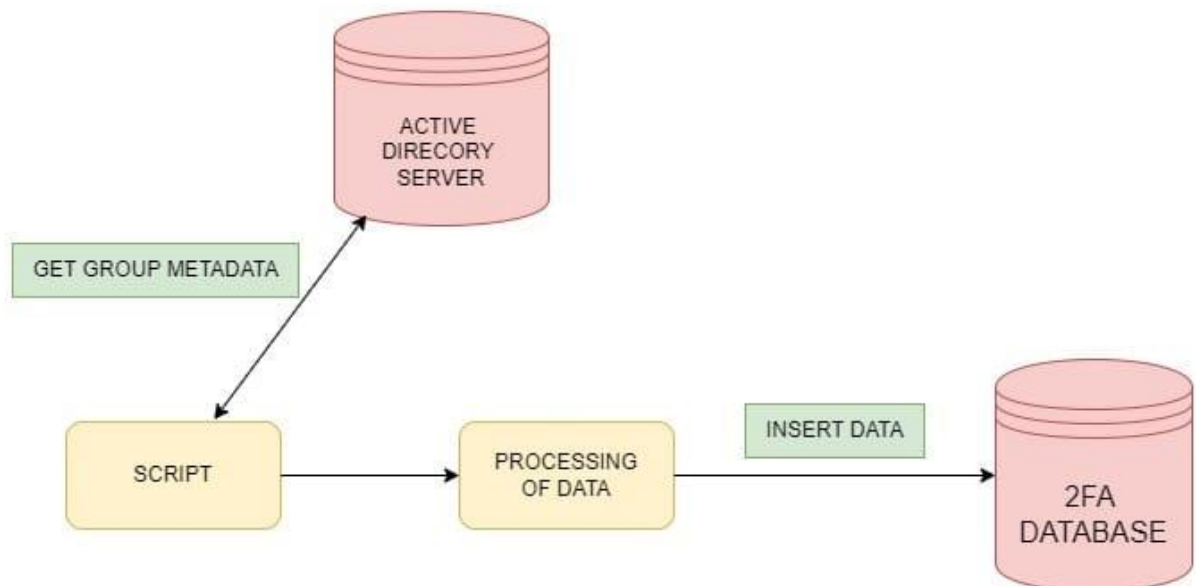
The script will run on a scheduled basis (e.g., daily) to ensure that the adding date of all users in the AD group is updated in the 2FA database.

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	GROUP_ID	NUMBER(38,0)	No	(null)	1	(null)
2	EMPLOYEE_ID	VARCHAR2(100 CHAR)	No	(null)	2	(null)
3	ADDED_DATE	DATE	No	(null)	3	(null)
4	NAME	VARCHAR2(50 CHAR)	No	(null)	4	(null)
5	EMAIL	VARCHAR2(100 CHAR)	No	(null)	5	(null)

Structure Of Member Added Date Table

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	GROUP_ID	NUMBER(38,0)	No	(null)	1	(null)
2	GROUP_NAME	VARCHAR2(100 CHAR)	No	(null)	2	(null)

Structure Of 2FA Group Name Table

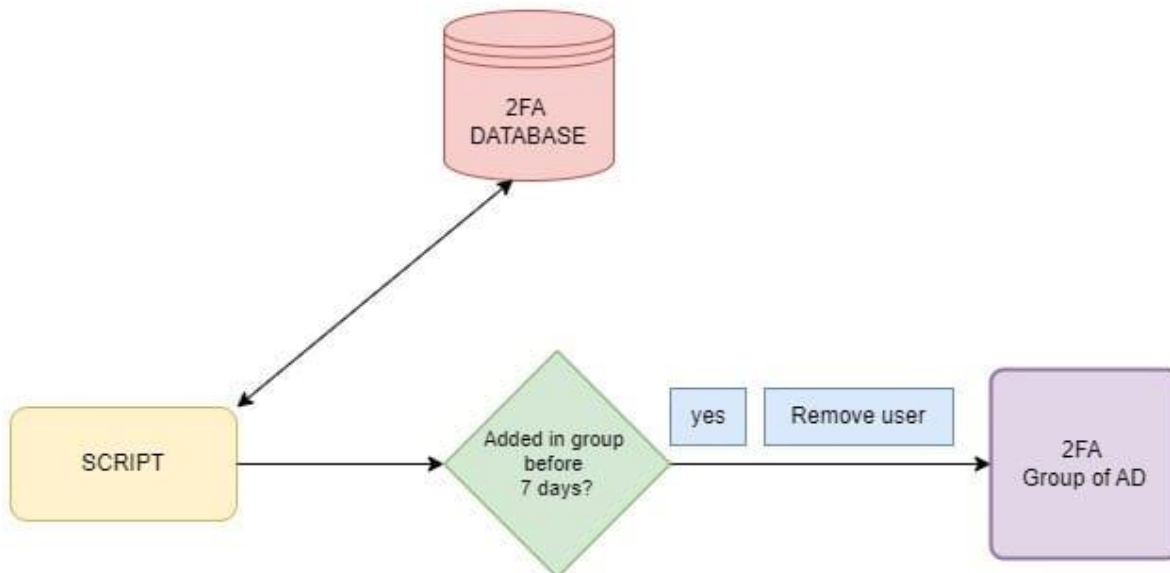


Flowchart Of Inserting Adding Date of Group Members

Step2:

Removing Users from AD Group Based on 2FA Database (Unix Script):

- The Unix script will query the 2FA database to retrieve the adding date of each user in the AD group.
- For each user, the script will compare the adding date with the current date and calculate the number of days the user has been in the AD group.
- If the user has been in the AD group for more than seven days, the script will remove the user from the AD group.
- The script will run on a scheduled basis (e.g., daily) to ensure that users who have been in the AD group for more than seven days are promptly removed.



Flowchart Of Removing Members From Group

4.2 SELF-SERVICE PORTAL

4.2.1 Need Of Self-Service Portal:

The self-service portal will allow users who have been removed from the Active Directory (AD) group for 2FA setup to request access to be added back to the group. The portal will streamline the process for users to request access, reducing the need for manual intervention and improving the overall user experience.

4.2.2 Workflow for Self-Service Portal:

- ❖ User Authentication:

User accesses the self-service portal and logs in using their credentials (username and password).

- ❖ 2FA Self-Service Page:

After login, the user is directed to the 2FA self-service page, where they can manage their 2FA settings.

- ❖ Enter Employee ID:

User enters their employee ID in the provided field on the 2FA self-service page.

- ❖ API Call to Check 2FA Status:

A request is made to the backend API to check if the user is already a member of the 2FA group.

- ❖ Check 2FA Status:

Backend API checks if the user is already a member of the 2FA group. If the user is already a member, a message is displayed indicating that the user is already a member of the 2FA group and can proceed to set up 2FA.

❖ Direct User to Set Up 2FA:

If the user is already a member of the 2FA group, a message is displayed instructing the user to go to the passcode website and set up 2FA.

❖ Request Access to 2FA Group:

If the user is not already a member of the 2FA group, a "Request Access" button is displayed. If the user clicks the "Request Access" button, a request is made to the backend API to add the user to the 2FA group.

❖ API Call to Add User to 2FA Group:

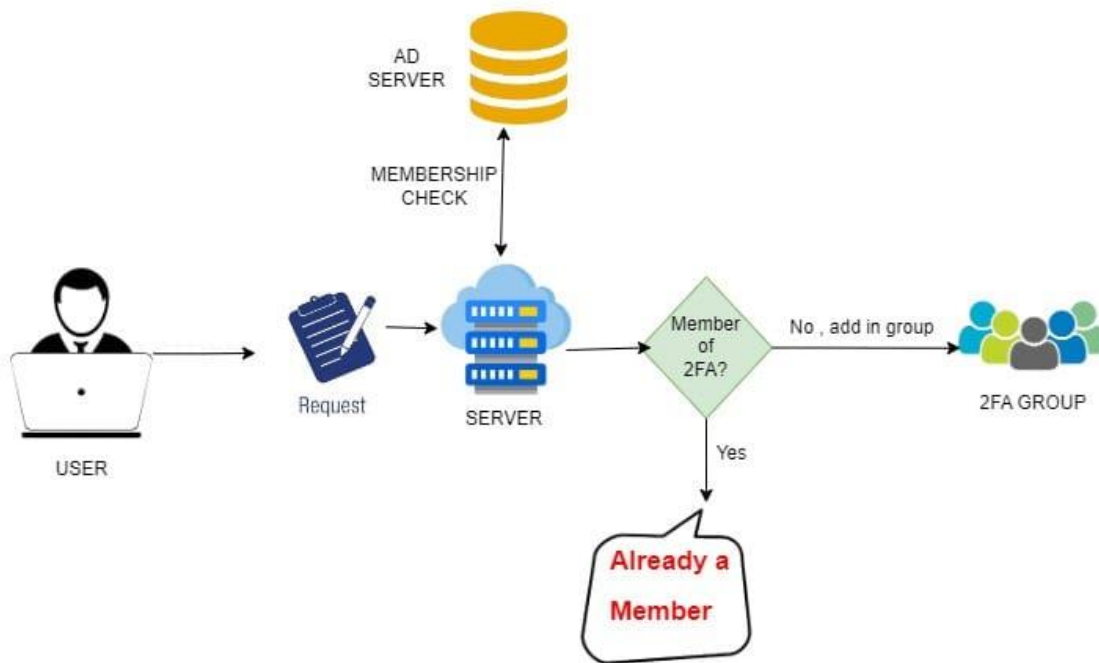
A request is made to the backend API to add the user to the 2FA group.

❖ User Added to 2FA Group:

Upon successful addition to the 2FA group, a message is displayed indicating that the user has been added to the 2FA group and can proceed to set up 2FA.

❖ Set Up 2FA:

The user is directed to the passcode website to set up 2FA.



Flowchart Of Self-Service Portal

4.2.3 Flask Api Endpoints:

There are two API endpoints. One is to get user status, which takes an employee ID and gives its status, such as whether the employee is a 2FA user or not or is added to the 2FA group.

Another one is to add to the 2FA group, which takes an employee ID and adds it to the 2FA group of active directory groups.

The screenshot shows a web browser window with the address bar displaying 'localhost:5000/api/v1/ui'. The page title is '2fa self service portal'. The main content area displays the Swagger UI for the 'self_service_2fa' API. The selected endpoint is 'GET /self_service_2fa/get_user_status'. The 'Parameters' section is empty. The 'Responses' section shows a table with status codes and descriptions:

Code	Description
200	get user status successful.
500	Internal server error.

Get User Status API endpoint

The screenshot shows a web browser window with the address bar displaying 'localhost:5000/api/v1/ui'. The page title is '2fa self service portal'. The main content area displays the Swagger UI for the 'self_service_2fa' API. The selected endpoint is 'POST /self_service_2fa/add_to_2fa_group'. The 'Parameters' section is empty. The 'Responses' section shows a table with status codes and descriptions:

Code	Description
200	get user status successful.
500	Internal server error.

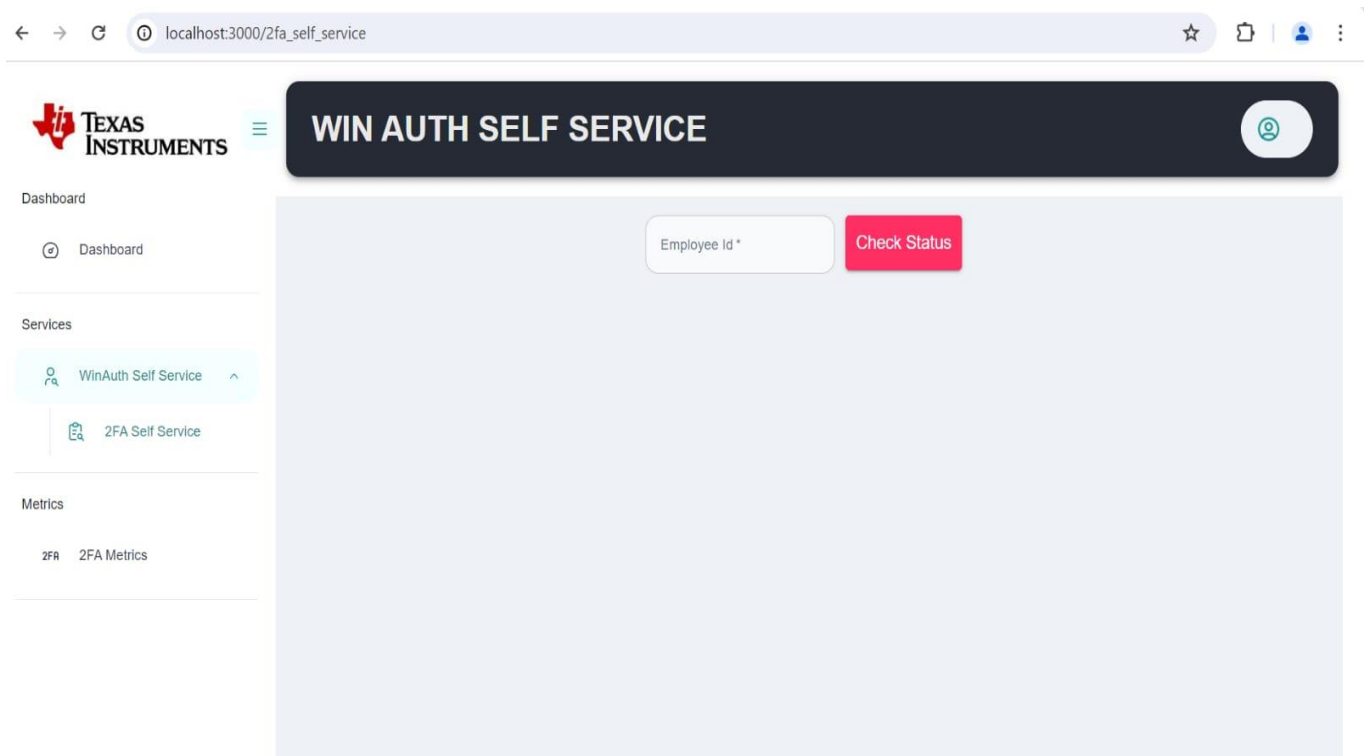
Add To 2FA Group API Endpoint

4.2.4 Types Of View:

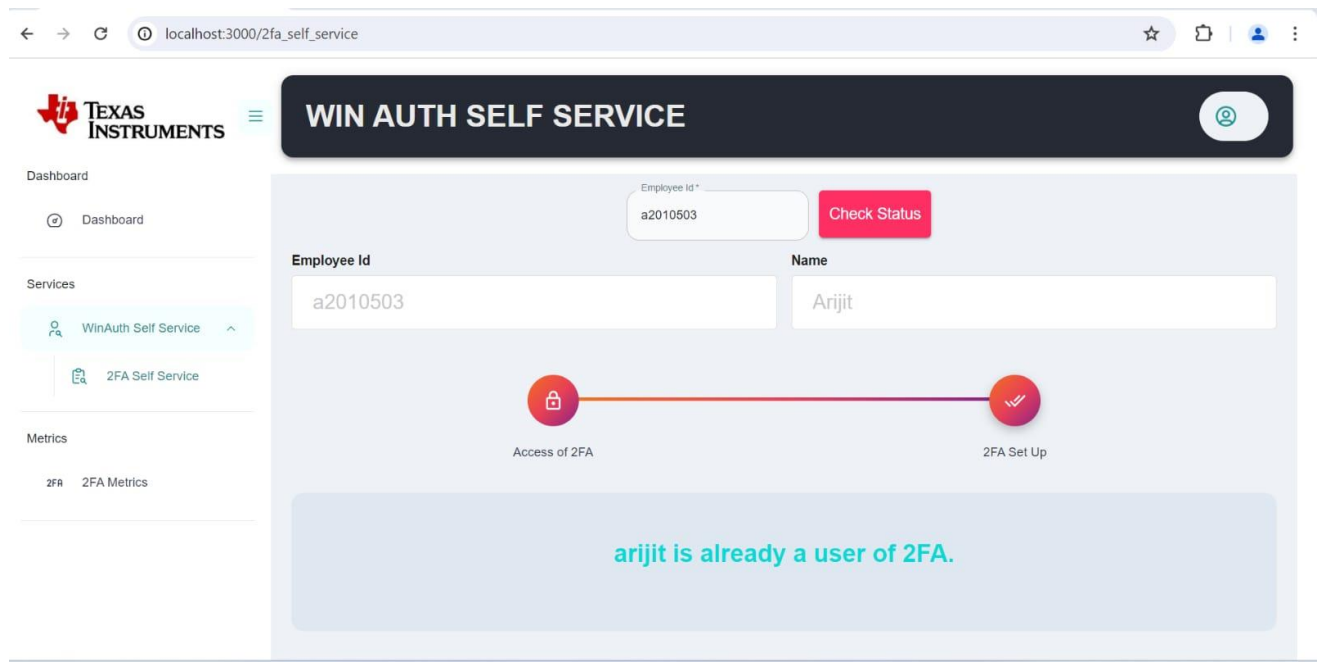
There is a landing page where users have to provide an employee ID.

There are three possibilities—

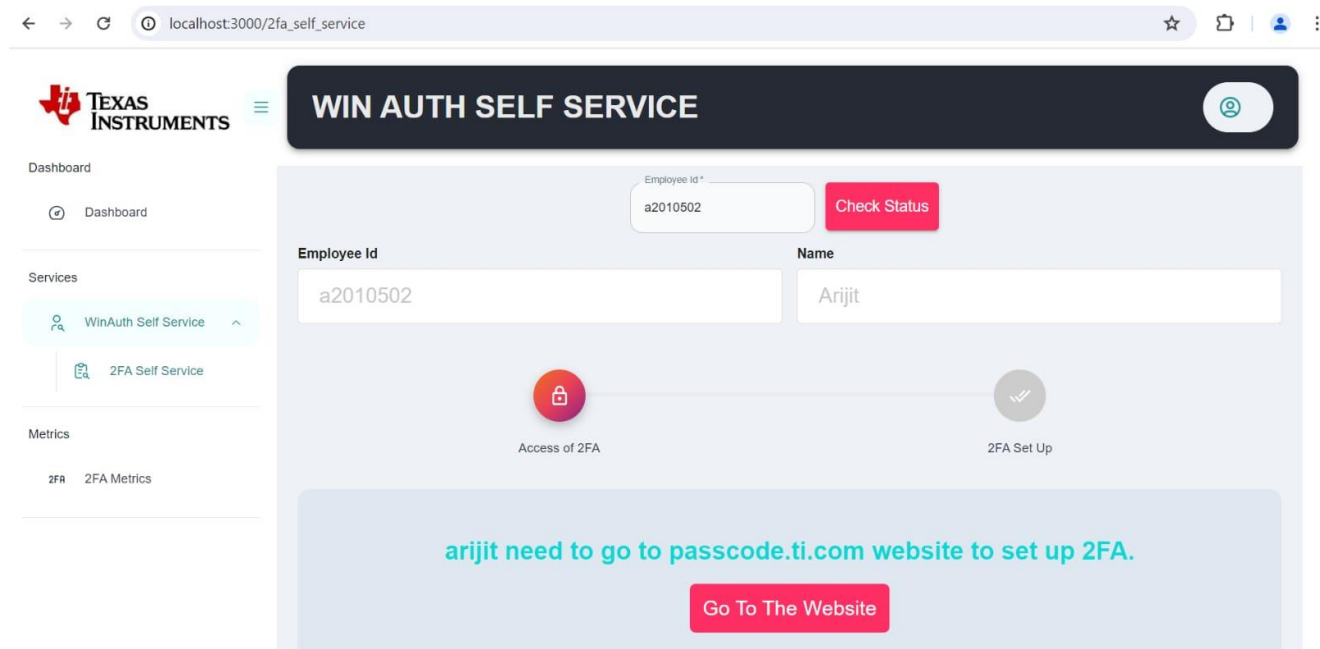
- ❖ User is already a 2FA user.
- ❖ User is a member of the 2FA group but has not set up 2FA.
- ❖ User is not a member of the 2FA group and needs the access.



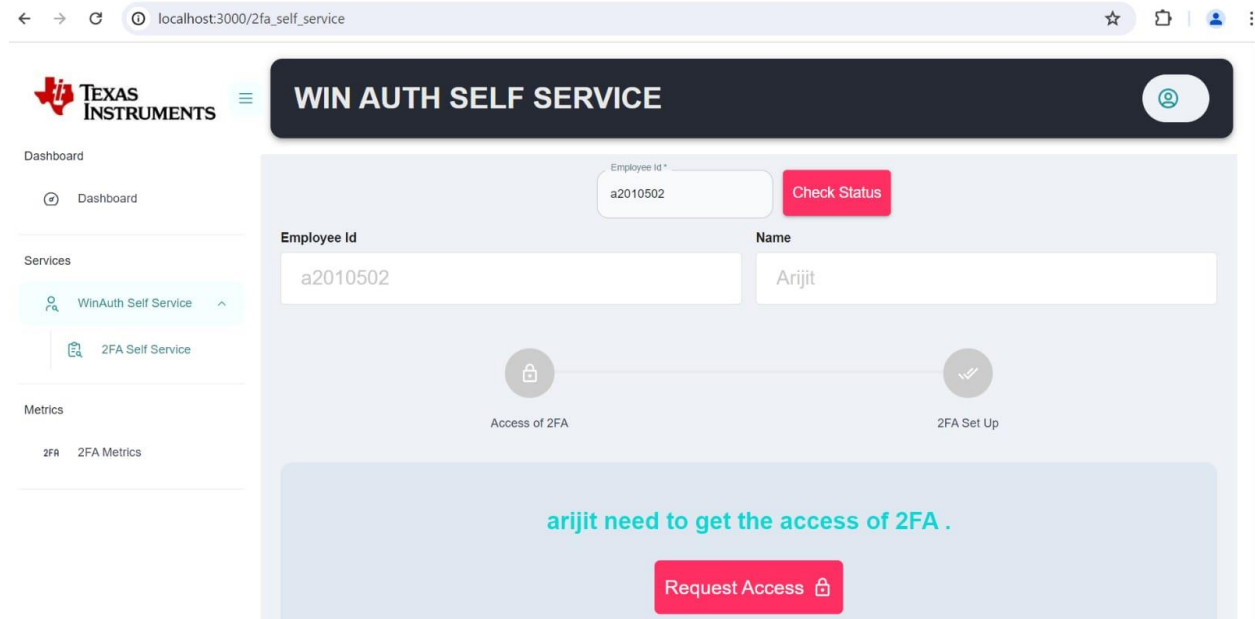
Landing Page view



Already 2FA User View



Already Member Of The Group View



Not A Member Of The Group View

4.3 Extracting Metrics

4.3.1 Need Of Metrics:

In the context of a Two-Factor Authentication (2FA) system, the extraction of metrics plays a crucial role in monitoring and improving the effectiveness of the security mechanism. Here are some key reasons highlighting the need for extracting metrics:

1. **Security Monitoring:** Metrics such as the number of users using different authentication methods (e.g., egrid, SMS, Entrust app), login failures, and locked-out smart card users provide insights into the security posture of the organization. Monitoring these metrics helps identify potential security threats and take proactive measures to mitigate them.
2. **User Experience Improvement:** Metrics related to incomplete user registrations and stale users help in understanding user behavior and experience. By analyzing these metrics, organizations can identify and address issues that may be causing inconvenience to users, thereby improving the overall user experience.
3. **Compliance and Audit Requirements:** Many industries and organizations have specific compliance and audit requirements related to security. Extracting metrics helps demonstrate compliance with these requirements by providing data on 2FA usage and effectiveness.
4. **Resource Optimization:** Metrics such as the number of active registered users and users holding multiple authentication methods help in optimizing resources. Organizations can allocate resources more efficiently based on the usage patterns and requirements indicated by these metrics.
5. **Identification of Trends and Patterns:** Organizations can identify trends and patterns in 2FA usage by analyzing metrics over time. This information can be used to predict future requirements and make informed decisions about the 2FA system.
6. **Risk Management:** Metrics related to non-compliant 2FA servers and smart cards about to expire help in managing risks associated with the 2FA

system. By addressing these issues proactively, organizations can reduce the likelihood of security breaches and other risks.

4.3.2 Workflow:

The extraction of metrics related to 2FA usage, login failures, incomplete registrations, and other aspects involves querying Active Directory (AD) groups for user information and processing this data to extract the desired metrics. To improve efficiency, the extraction process utilizes multiprocessing to handle a large number of users.

1. Querying Active Directory Groups:

- The script starts by querying the AD groups to retrieve the groups' distinguished names (DN) containing the users for whom metrics need to be extracted.

2. Retrieving User Details:

- For each distinguished name obtained from the AD groups, the script retrieves the details of each user in the group.
- The user details include authentication methods (e.g., egrid, SMS, Entrust app), smart card status, and registration status.

3. Extracting Metrics:

- Based on the user details, the script extracts the following metrics:
 - Number of users using egrid for 2FA (instant-weekly).
 - Number of login failures vs. success (bad 2FA) for egrid, SMS, and Entrust app (per quarter).
 - Number of incomplete user registrations (instant).
 - Number of stale users (per quarter).
 - Number of users holding more than one authentication method (instant).

- Number of locked-out smart card users.
- The number of smart cards is about to expire next month (auto-email).
- Number of unique users using SMS (per quarter).
- Number of active registered users (license).
- Number of non-compliant 2FA servers.

4. Multiprocessing:

- The extraction process uses multiprocessing to handle the large number of users (approximately 36,000 employees).
- Multiprocessing allows the script to run multiple processes simultaneously, improving data extraction speed.
- The script is configured to run up to 20 processes at a time, optimizing the extraction process.

5. Inserting Metrics into Database:

- Once the metrics are extracted, they are inserted into a metrics database for storage and analysis.
- The metrics database stores the extracted metrics in a structured format for easy retrieval and display.

6. Displaying Metrics on Web Page:

- The extracted metrics are displayed on a web page, allowing administrators to view and analyze the 2FA usage and performance metrics.
- The web page provides a user-friendly interface for monitoring and managing the 2FA system.

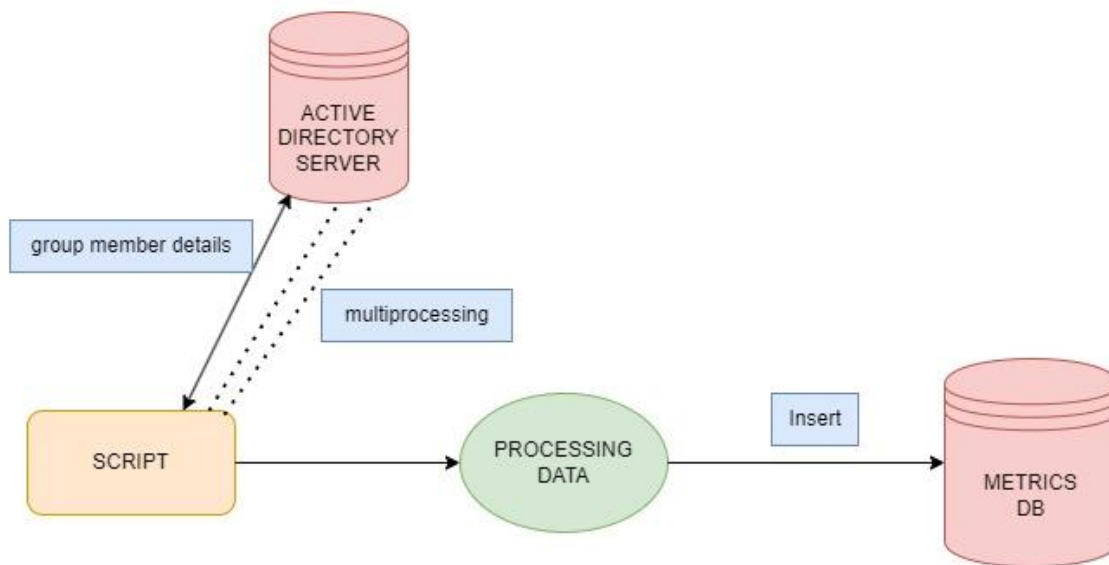


Diagram Of Metrics Script

4.3.3 Endpoints:

[Base URL: /api/v1]
/api/v1/swagger.json

Welcome to the 2fa self service portal API documentation site!

Authorize

metrics_info

- POST** /metrics_info/get_available_dates_of_a_metric Give date lists in which metric have stored in DB
- POST** /metrics_info/get_employee_details_of_metric_of_specific_date Give Employee details of a specific metric of a date
- POST** /metrics_info/get_metric_detailed_data_in_email Give Metric detailed data in email
- GET** /metrics_info/get_metric_difference_date_choosing_options Give the dates in which metric data are there in Database
- POST** /metrics_info/get_metrics_name_and_latest_value Give Metric Names and value and difference between today and a specific date

4.3.4 Database Design:

Metric_names
metric_id(pk)
metric_name

Group_names
group_id(pk)
group_name

s

Metrics_Data
metric_data_id(pk)
group_id(fk)
metric_id(fk)
metric_value
date_time

Employee_Details
Employee_id(pk)
Name
Email_id

Metric_Data_ID_Employee_Mapping	Pk—combination Of two
Metric_data_id(fk)	
Employee_id(fk)	

Chapter 5. Deployment

Docker Containers:

- The application is deployed using Docker containers.
- Separate containers are used for the frontend (React) and backend (Flask) components to ensure modularity and ease of maintenance.

CI/CD with Jenkins:

- Jenkins is used for Continuous Integration and Continuous Deployment (CI/CD).
- Jenkins pipelines are configured to automate the build, test, and deployment processes.
- The pipeline includes stages for building Docker images, running tests, and deploying the application to the production environment.

Automatic Adding and Removal Scripts:

- Scripts for automatically adding and removing users from the 2FA group are scheduled in the crontab of a Unix machine.
- These scripts are executed daily to ensure that users are added or removed from the 2FA group per the defined criteria.

Deployment Process:

1. **Code Commit:** Developers commit code changes to the version control system (e.g., Git).
2. **Trigger Jenkins Build:** The code commit triggers a Jenkins build job.
3. **Build Docker Images:** Jenkins builds Docker images for the frontend and backend components.
4. **Run Tests:** Automated tests are run to ensure the quality and stability of the code.

5. **Deploy to Staging:** If the tests pass, the application is deployed to a staging environment for further testing.
6. **Manual Approval:** After testing in the staging environment, a manual approval step is required to deploy the application to the production environment.
7. **Deploy to Production:** Jenkins deploys the application to the production environment upon approval.

Benefits:

- Docker containers ensure consistency between development, testing, and production environments.
- Jenkins automates the build, test, and deployment processes, reducing the risk of errors and improving efficiency.
- Automatic adding and removal scripts ensure user access to 2FA is managed automatically, reducing manual intervention and improving security.

Chapter 6. Result And Analysis

Automation of 2FA Registration Process:

- Automating the 2FA registration process has significantly reduced the time and effort required for users to set up 2FA.
- The process has become more streamlined and user-friendly by automatically adding users to the 2FA group based on the completion of training.
- The use of PowerShell scripts and multiprocessing has ensured that the process is efficient and can handle a large number of users.

Extraction of Metrics:

- The extraction of metrics related to 2FA usage, login failures, and user registration has provided valuable insights into the effectiveness of the 2FA system.
- Metrics such as the number of users using egrid for 2FA, login failures vs. success, and incomplete user registrations have helped identify areas for improvement.
- The use of a metrics database and web page for displaying the metrics has made it easier for administrators to monitor and analyze the 2FA system.

Impact on Security and User Experience:

- The automation of the 2FA registration process and the extraction of metrics have positively impacted security and user experience.
- The system has become more secure and user-friendly by simplifying the registration process and providing administrators with valuable insights.
- Users now have a more efficient way to set up 2FA, and administrators can proactively identify and address security issues.

Chapter 7. Conclusion And Scope Of Future Work

Implementing the automated 2FA registration process, extracting metrics related to 2FA usage, and developing the self-service portal have significantly improved the security, efficiency, and user experience of the 2FA system. By leveraging technologies such as Flask, React, PowerShell script, Docker, and Jenkins, we have successfully automated the 2FA registration process, extracted key metrics for monitoring and analysis, and provided users with a convenient way to manage their 2FA settings.

While the current implementation has achieved significant improvements, there are several areas for future work and enhancement:

1. **Enhanced Metrics Analysis:** Develop more advanced analytics and visualization tools to gain deeper insights into 2FA usage patterns and security trends.
2. **Integration with Third-Party Services:** Explore integration with third-party services for additional security features or authentication methods, such as biometric authentication.
3. **Enhanced User Experience:** Continuously improve the self-service portal interface and user experience based on user feedback and usability studies.
4. **Scalability and Performance:** Optimize the system for scalability and performance to handle increasing numbers of users and data.
5. **Security Audits and Compliance:** Conduct regular security audits and ensure compliance with relevant standards and regulations to maintain a high level of security.
6. **Mobile Application Development:** Develop a mobile application for the self-service portal to provide users with more flexibility and convenience in managing their 2FA settings.
7. **Integration with Other Systems:** Explore integration with other systems within the organization to enhance security and streamline authentication processes.

By addressing these areas of future work, we can further enhance the security, efficiency, and user experience of the 2FA system, ensuring that it remains a robust and reliable security solution for the organization.

Chapter 8. References

1. Flask:
 - Flask Documentation: [**https://flask.palletsprojects.com/en/2.0.x/**](https://flask.palletsprojects.com/en/2.0.x/)
 - Flask Tutorial: [**https://flask.palletsprojects.com/en/2.0.x/tutorial/**](https://flask.palletsprojects.com/en/2.0.x/tutorial/)
2. React:
 - React Documentation: [**https://reactjs.org/docs/getting-started.html**](https://reactjs.org/docs/getting-started.html)
 - React Tutorial: [**https://reactjs.org/tutorial/tutorial.html**](https://reactjs.org/tutorial/tutorial.html)
3. PowerShell Scripting:
 - Microsoft Docs - PowerShell: [**https://docs.microsoft.com/en-us/powershell/**](https://docs.microsoft.com/en-us/powershell/)
4. 2FA (Two-Factor Authentication):
 - What is Two-Factor Authentication (2FA)?:
[**https://www.cisco.com/c/en/us/products/security/two-factor-authentication-2fa-what-is-it.html**](https://www.cisco.com/c/en/us/products/security/two-factor-authentication-2fa-what-is-it.html)
 - Implementing Two-Factor Authentication Using Time-Based One-Time Passwords: [**https://www.ietf.org/rfc/rfc6238.txt**](https://www.ietf.org/rfc/rfc6238.txt)
5. Docker:
 - Docker Documentation: [**https://docs.docker.com/**](https://docs.docker.com/)
 - Docker Overview: [**https://www.docker.com/why-docker**](https://www.docker.com/why-docker)
6. Jenkins:
 - Jenkins Documentation: [**https://www.jenkins.io/doc/**](https://www.jenkins.io/doc/)