|  |  |
| --- | --- |
| **Name** : Muhammad Zain-Ul-Abideen | **Domain** : Cyber Security |
| **CID** : DEP6384 | **Task Instructor** : Noor-Us-Sama |
| **Topic** : Multi-Factor Authentication (MFA) | **Tools** : Google Authenticator & CMD Terminal |

**Digital Empowerment Network Task 2**

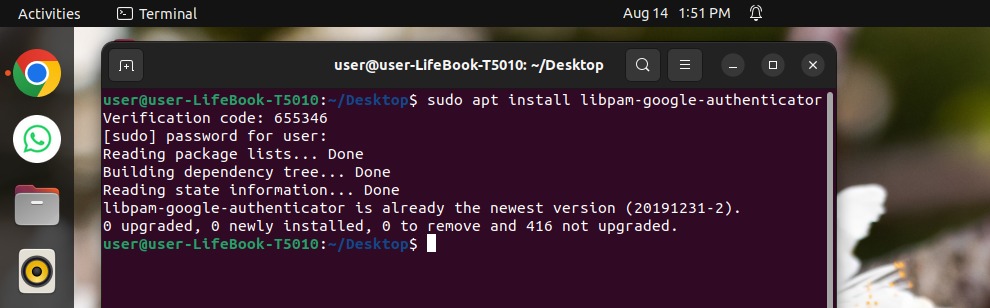
**1. Objective: Increase account security by implementing MFA.**

**2. Description: Set up multi-factor authentication (MFA) to add an extra layer of security to user accounts. Ensure MFA is user- friendly and effective.**

**3. Key Steps:**

* **Selecting an MFA solution compatible with the system.**
* **Configuring MFA settings and options.**
* **Educating users on how to set up and use MFA.**
* **Monitoring MFA adoption and addressing any issues.**
* **Regularly updating and maintaining MFA configurations.**

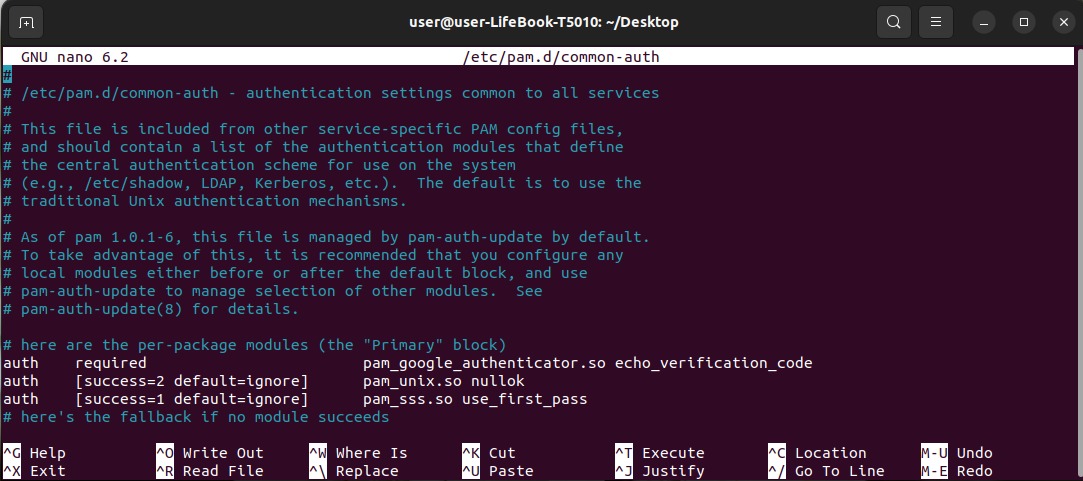
**Answer** : **Multi-Factor Authentication (MFA)** is necessary because it adds an extra layer of security by requiring multiple forms of verification (e.g., **password + a code sent to your phone**). This reduces the risk of unauthorized access, even if one factor (like a password) is compromised. MFA protects against common threats like **phishing**, **password theft**, and **brute-force attacks**, making it much harder for attackers to gain access to sensitive accounts or data. So we are practically going to implement **Multi-Factor Authentication (MFA)** for our system. For the above given task, I’m going to use **Google Authenticator** in collaboration with **Command Terminal** as it is compatible with the system and I’m implementing this on **Ubuntu 22.04 LTS ( Jammy Version )** instead of a **VMWare.** So first thing is first we have to download the **Google Authenticator** application on **Ubuntu.** It can be download by using the following commands in **Terminal** like given as follows.



Once the **Google Authenticator** has been download you can start to configure the settings for it using a text editor and the following command, I’m using **Nano** but you can use your default editor.

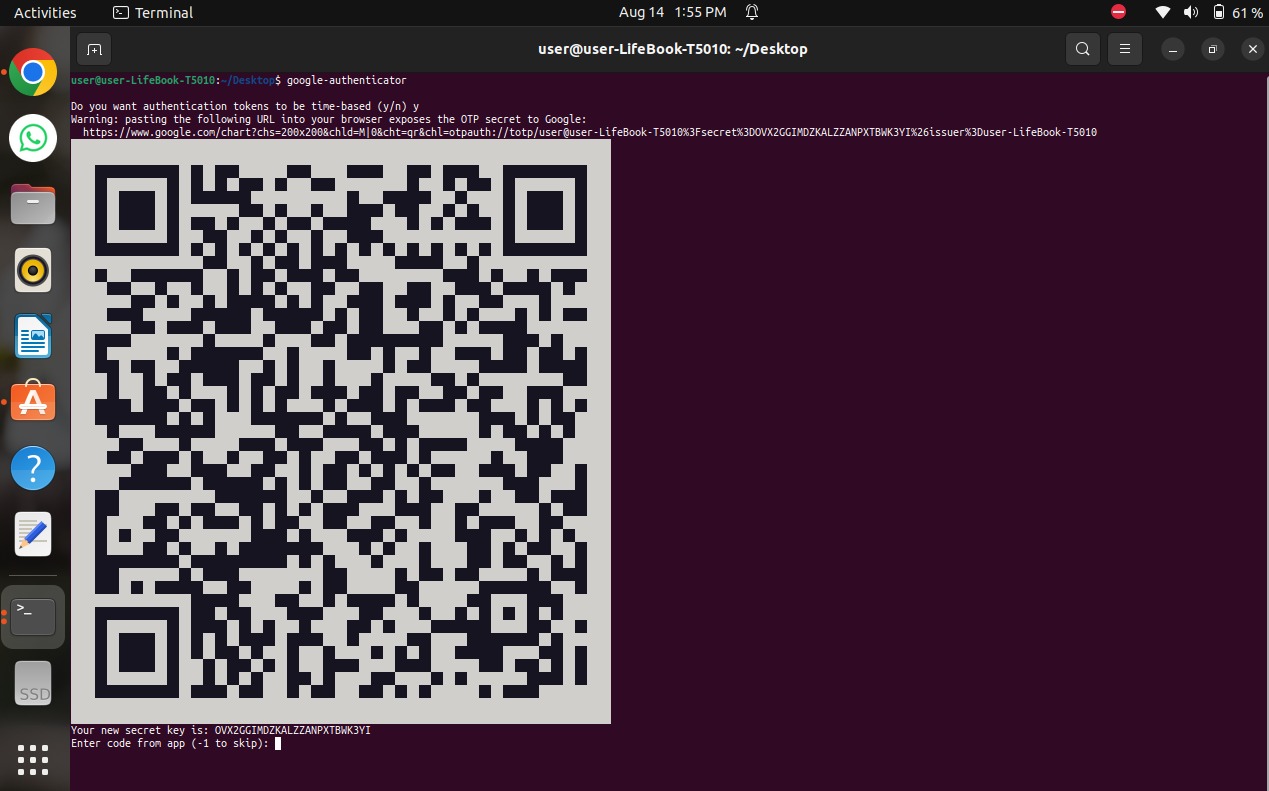
****

Running this command in the terminal will open up the **Configuration Settings** of the **Google Authenticator** where we need to make a few changes to enable the **Multi-Factor Authentication (MFA).**

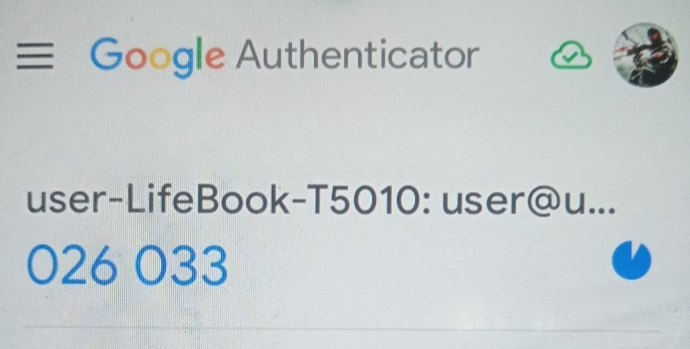
****



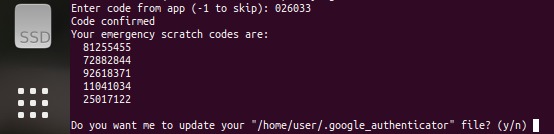
In the **Primary Block** section we need to edit and add a new line which will follow with **auth**, **required**, **pam\_google\_authenticator.so echo\_verification\_code.** This will enable the **Google Authenticator.**



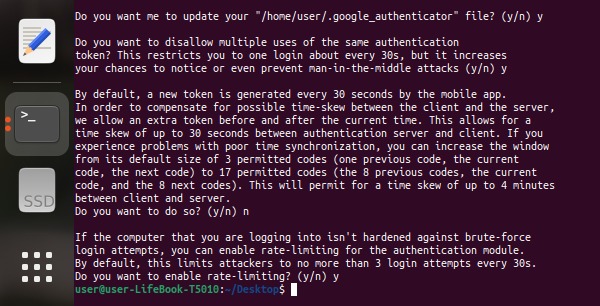
Now we will download **Google Authenticator Android Application** on a **Android Mobile Device** so we can successfully start to use the **Google Authenticator** to protect our device. But first we have to sync the **Mobile Application** & the one **Google Authenticator** in the system. So we open the **Android Application** , we press the **“Plus” icon** which will give option of **Adding Code.** Once we click on it , it will give option of a **“Scan A QR Code”** then scan the given **QR Code** on the **Terminal** .This will sync both of the device and the **Mobile Application** will be getting codes in intervals of **30 Seconds**. We will receive the code on the **Mobile Device** as follows.



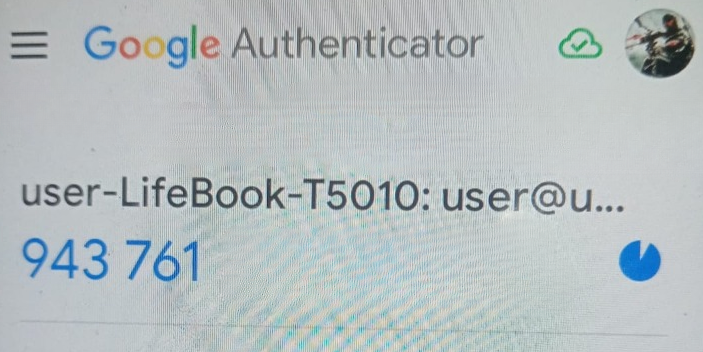
Which we will have to add as our **Confirmation Code** in the **Terminal.**



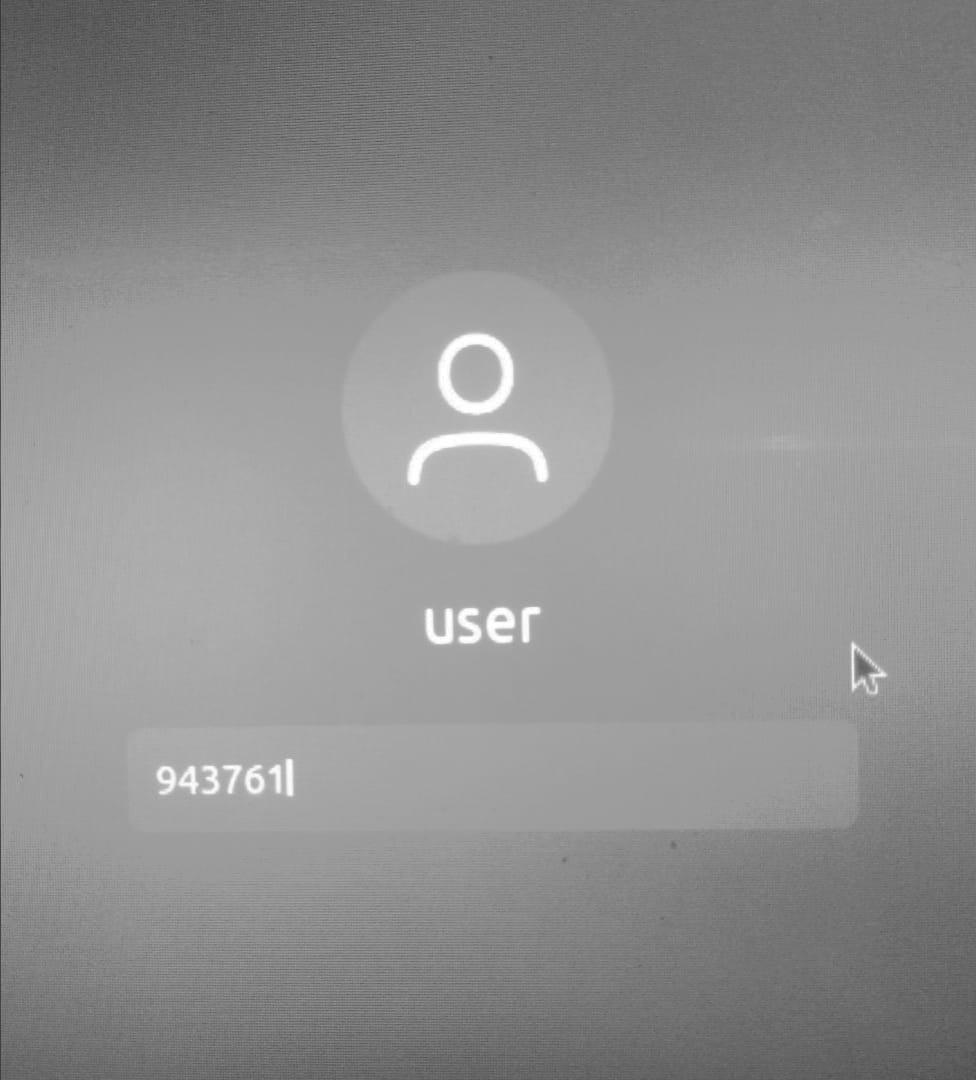
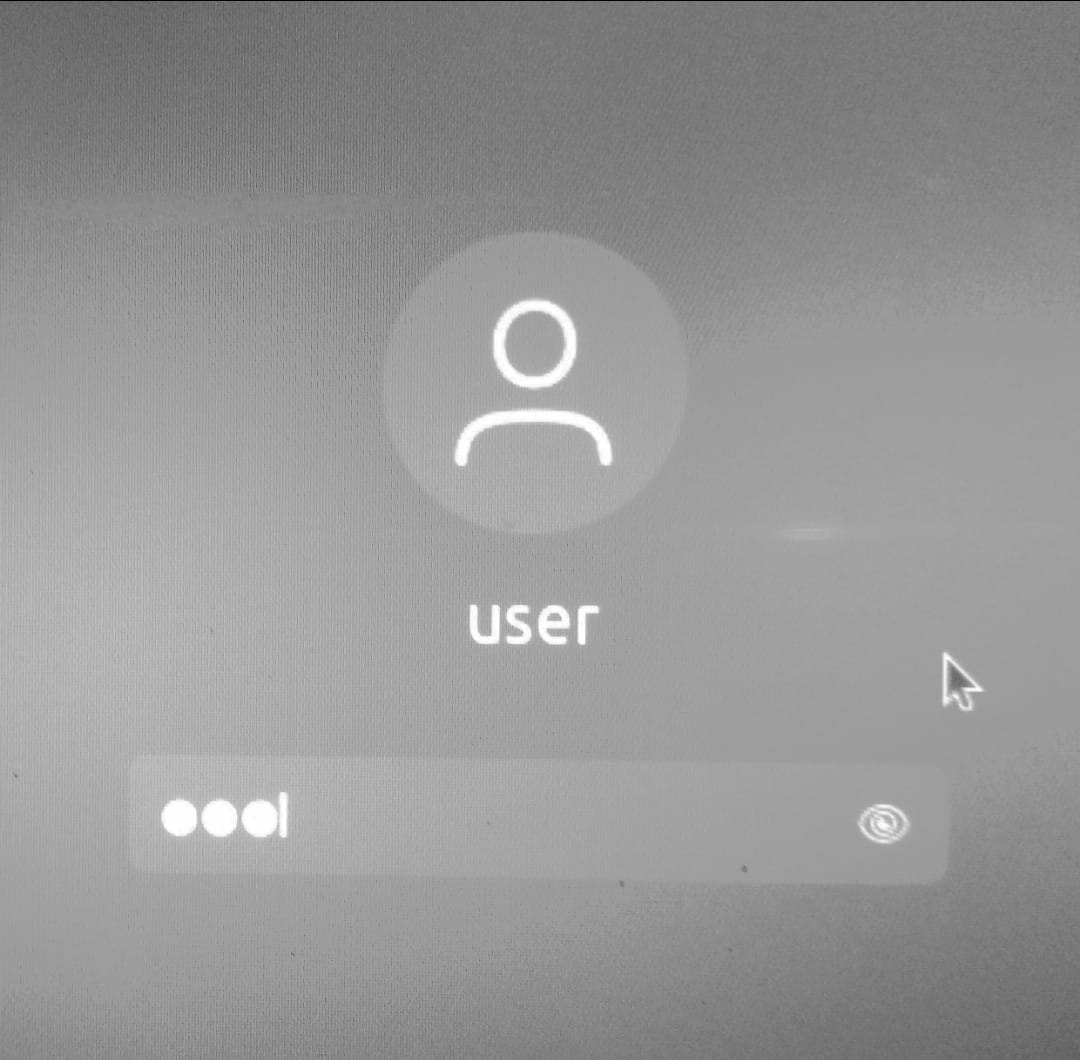
Upon receiving the **Confirmation Code** a message of **Code Confirmed** will be displayed on the **Terminal .** You will also be provided with a list of **Emergency Scratch Codes.**  As well as acknowledgement to update the **Google Authenticator File** which you will have to respond with a **Y** if code is confirmed. It will then ask for further **Preferences** before completely launching the **Google Authentication.** Here you can select the desired option in example the **Time of Interval** before **Verification Code** gets changed or preventions from different types of attacks that can occur on the system with special regards and prevention for **Man-In-The-Middle Attacks**  with respect to a lot of different types of attacks too. Other **Preferences Options** may involve **Time Skewing** And **Rate Limiting**  as observed in the screen below.



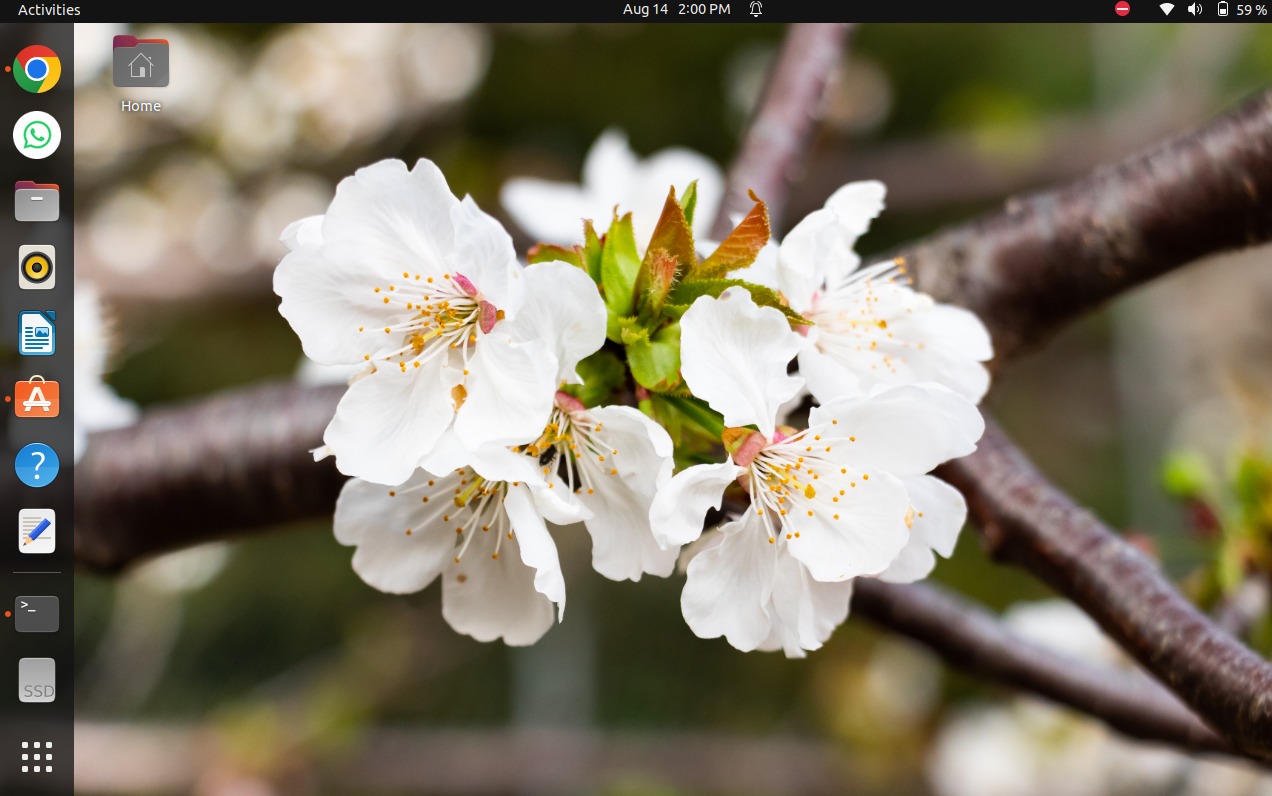
Once your done with selecting the desired options and **Preferences,** the **Google Authenticator**  will be effective on your device and will implement **Multi-Factor Authentication (MFA)**, it adds an extra layer of security by requiring multiple forms of verification on your system. Now we will **Lock**  the screen of the system which normally brings us to the **Login Panel** where only authentication method shown is to enter your **System’s Password.** But this time it asks for a **Verification Code** before hand even before the **Password**. The **Verification Code**  is sent to the **Google Authenticator Mobile Application.** It will stay for **30 Seconds** before changing into a new **Verification Code** .



We copy code from the **Application** and enter it in the **Verification Code Authentication Panel,** which will check if the **Verification Code** matches the one on the **Application.** If they both match user is then and only then sent to the **System Password Authentication Panel** , where He/She will enter password of the system and system will check again if the password matches to the actual **System’s Password** ensuring a **Double Layer Of Protection** for the system. The process follows like this :

After both the **Verification Code** & **System’s Password** is entered correctly and successfully, the **User** is granted access to the system which means only authorized user can use this system and it is well protected by **Threats** and any **Unauthorized** or **Unwanted Attention**.



If a user however wants to disable the **Google Authentication System**, it is easy to do so we type the command “**sudo nano /etc/pam.d/common-auth**” and we comment the line where we made changes in the first time configurations observed in the **Fourth Screen**  where we wrote **auth**, **required**, **pam\_google\_authenticator.so echo\_verification\_code** , we do it like so :



We change this first line into a comment by putting a **#** before starting the line like so :



This will disable the **Google Authenticator,** but it is recommended to enable it just so that system is kept protected in regards to **Multi-Factor Authentication (MFA**), as it crucial because it verifies a user's identity through multiple independent factors, making unauthorized access significantly harder. Even if one factor, like a password, is compromised, additional steps such as a fingerprint or a one-time code are required, providing a stronger defense against cyber threats. This layered security approach is essential for protecting sensitive information and reducing the likelihood of breaches.