DIGITAL EMPOWERMENT NETWORK

INTERNSHIP REPORT

WEEK#02

PRESENTED BY ALIZA JAVED
CID DEN10161

CONTENTS

ABOUT THE COMPANY

At Digital Empowerment Network, we are dedicated to empowering Pakistan's university students through a wide range of initiatives that foster leadership, academic growth, and technical expertise. By bridging the digital divide with workshops, coding camps, and hackathons, we equip students with essential skills to thrive in the modern workforce. Our commitment to education extends beyond the classroom, as we reinvest proceeds from our revenue-generating partnerships into impactful programs and charitable initiatives, providing resources and support to underserved communities. We believe in the power of education to create well-rounded individuals who can drive innovation and make a positive impact on society..

IMPLEMENTING MULTI-FACTOR AUTHENTICATION (MFA)

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.

INTRODUCTION TO MULTI-FACTOR AUTHENTICATION (MFA)

Multi-Factor Authentication (MFA) is a security mechanism that requires users to provide multiple forms of verification to gain access to a system, application, or network. The core idea behind MFA is to enhance security by ensuring that an individual is who they claim to be through the combination of two or more independent credentials: something the user knows (a password or PIN), something the user has (a token or smart card), and something the user is (a biometric verification like fingerprint or facial recognition).

In traditional authentication systems, the use of a single factor, usually a password, is the primary method of securing accounts. However, this single layer of protection has proven to be increasingly vulnerable to threats such as phishing attacks, password theft, and brute-force attempts. With the rise of sophisticated cyberattacks, MFA provides an extra layer of defense that reduces the risk of unauthorized access by requiring additional evidence of identity. **MFA** works by combining different types of authentication factors that fall into three categories:

1. **Knowledge Factors** (something you know): This typically refers to a password, PIN, or answers to security questions.
2. **Possession Factors** (something you have): This could be a physical token, a mobile device that generates one-time passcodes (OTP), or a smart card.
3. **Inherence Factors** (something you are): These are biometrics, such as a fingerprint, facial recognition, voice recognition, or retinal scan.

When a user attempts to log in to an MFA-enabled system, they will be prompted for additional verification beyond just a username and password. For example, after entering their password, they may receive a text message with a one-time code to their phone, or they may need to scan their fingerprint using a biometric reader. This layered approach increases the difficulty for attackers, as even if one factor (like a password) is compromised, the attacker would still need access to the second factor to complete the login process.

WHY USE MULTI-FACTOR AUTHENTICATION (MFA)

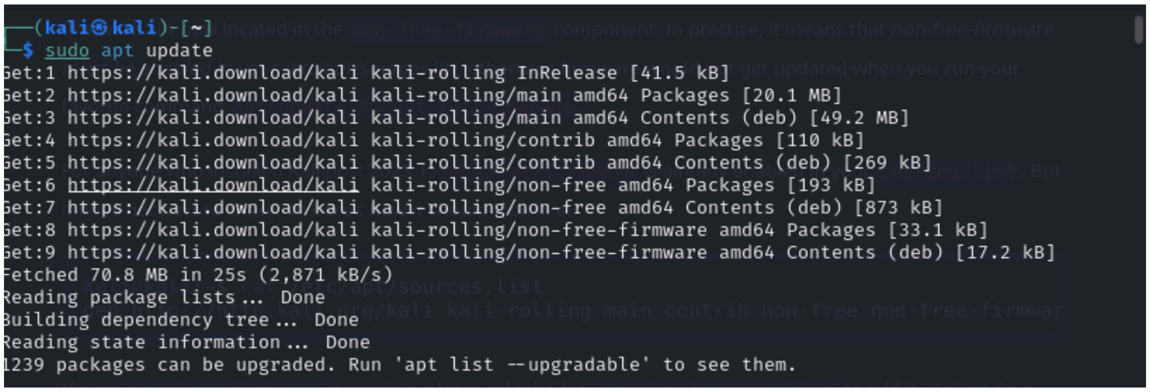
The importance of MFA in today's digital landscape cannot be overstated. As more businesses and individuals rely on online services for everything from communication to financial transactions, the need for stronger security measures becomes essential. Cybercriminals have adapted their tactics over time, finding new ways to exploit weak security practices. Passwords, no matter how complex, are often insufficient on their own. According to various cybersecurity reports, many breaches can be attributed to weak, reused, or compromised passwords. MFA offers a solution by adding multiple barriers to entry, ensuring that even if one authentication factor is compromised, the account remains secure.

Furthermore, MFA is widely recognized as a best practice in both corporate and consumer environments. Organizations handling sensitive data, such as financial institutions, healthcare providers, and government agencies, are increasingly required to implement MFA by regulations and industry standards. Moreover, the rise of remote work and cloud computing has made MFA even more critical, as users are now accessing systems from various locations, often outside traditional security perimeters.

MFA also helps to improve user confidence. Knowing that their accounts are protected by more than just a password provides peace of mind and reassures individuals and businesses that their sensitive information is safe. While MFA may introduce a small amount of inconvenience in terms of extra steps during the login process, the benefits in terms of security far outweigh the drawbacks.

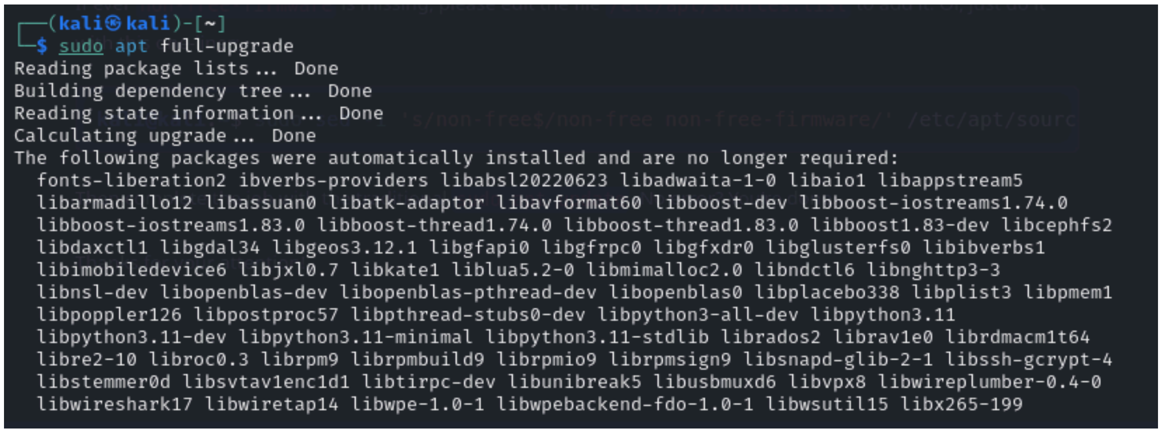
MFA IMPLEMENTATION USING GOOGLE AUTHENTICATOR

1. Install and configure google authenticator
    - Login to your kali linux
    - Open terminal
    - Before installing the google authenticator , it is good to update your system.



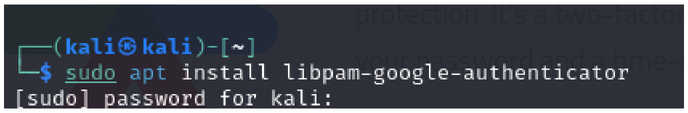
```
┌──(kali㉿kali)-[~]
└─$ sudo apt update
Get:1 https://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:2 https://kali.download/kali kali-rolling/main amd64 Packages [20.1 MB]
Get:3 https://kali.download/kali kali-rolling/main amd64 Contents (deb) [49.2 MB]
Get:4 https://kali.download/kali kali-rolling/contrib amd64 Packages [110 kB]
Get:5 https://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [269 kB]
Get:6 https://kali.download/kali kali-rolling/non-free amd64 Packages [193 kB]
Get:7 https://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [873 kB]
Get:8 https://kali.download/kali kali-rolling/non-free-firmware amd64 Packages [33.1 kB]
Get:9 https://kali.download/kali kali-rolling/non-free-firmware amd64 Contents (deb) [17.2 kB]
Fetched 70.8 MB in 25s (2,871 kB/s)
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
1239 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

- Upgrade to Include New Components



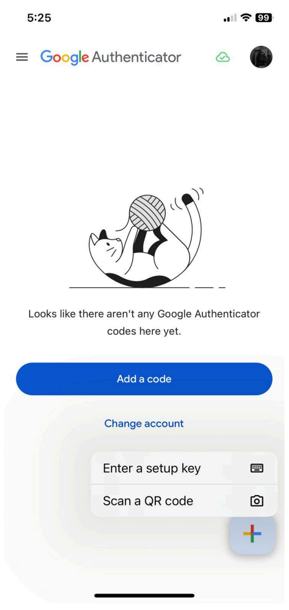
```
┌──(kali㉿kali)-[~]
└─$ sudo apt full-upgrade
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
Calculating upgrade ... Done
The following packages were automatically installed and are no longer required:
  fonts-liberation2 ibverbs-providers libabsl20220623 libadwaita-1-0 libaio1 libappstream5
  libarmadillo12 libassuan0 libatk-adaptor libavformat60 libboost-dev libboost-iostreams1.74.0
  libboost-iostreams1.83.0 libboost-thread1.74.0 libboost-thread1.83.0 libboost1.83-dev libcephfs2
  libdaxctl1 libgdal34 libgeos3.12.1 libgfapi0 libgfrpc0 libgfxdr0 libglusterfs0 libibverbs1
  libimobiledevice6 libjxl0.7 libkate1 liblua5.2-0 libmimalloc2.0 libndctl6 libnghttp3-3
  libnsl-dev libopenblas-dev libopenblas-pthread-dev libopenblas0 libplacebo338 libplist3 libpmem1
  libpoppler126 libpostproc57 libpthread-stubs0-dev libpython3-all-dev libpython3.11
  libpython3.11-dev libpython3.11-minimal libpython3.11-stdlib librados2 librav1e0 librdmacm1t64
  libre2-10 libroc0.3 librpm9 librpmbuild9 librpmio9 librpmsign9 libsnapd-glib-2-1 libssh-gcrypt-4
  libstemmer0d libsvtav1enc1d1 libtirpc-dev libunibreak5 libusbmuxd6 libvpx8 libwireplumber-0.4-0
  libwireshark17 libwiretap14 libwpe-1.0-1 libwpebackend-fdo-1.0-1 libwsutil15 libx265-199
```

- Install google authenticator.

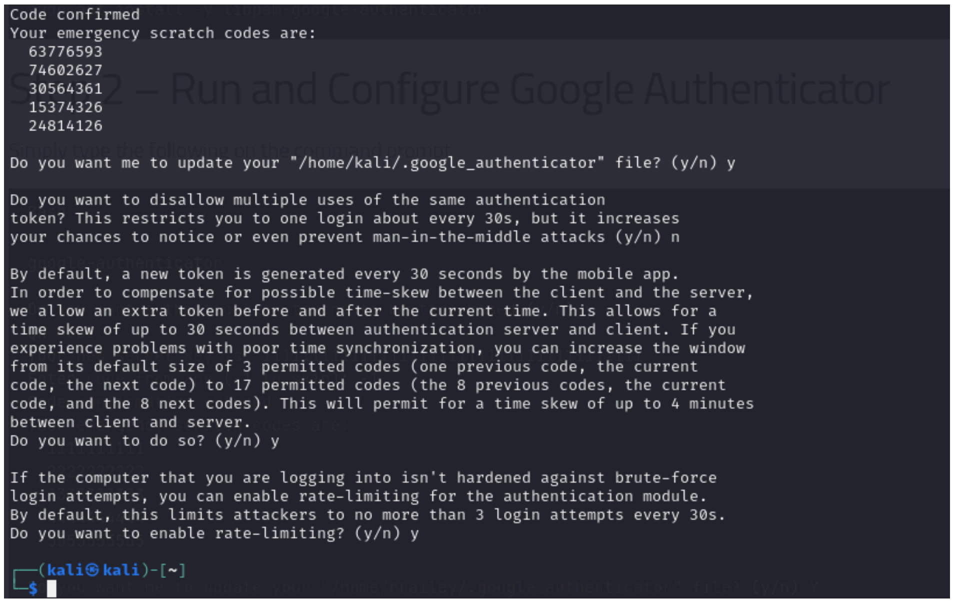


```
┌──(kali㉿kali)-[~]
└─$ sudo apt install libpam-google-authenticator
[sudo] password for kali:
```

- Download google authenticator in your phone and scan the qr code from your kali linux.



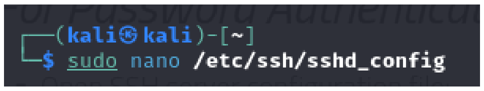
- Enter the secret key that appeared in your google authenticator app. Update the google authenticator configuration by answering 'y' to the remaining questions.



```
Code confirmed
Your emergency scratch codes are:
  63776593
  74602627
  30564361
  15374326
  24814126

Do you want me to update your "/home/kali/.google_authenticator" file? (y/n) y

Do you want to disallow multiple uses of the same authentication
token? This restricts you to one login about every 30s, but it increases
your chances to notice or even prevent man-in-the-middle attacks (y/n) n

By default, a new token is generated every 30 seconds by the mobile app.
In order to compensate for possible time-skew between the client and the server,
we allow an extra token before and after the current time. This allows for a
time skew of up to 30 seconds between authentication server and client. If you
experience problems with poor time synchronization, you can increase the window
from its default size of 3 permitted codes (one previous code, the current
code, the next code) to 17 permitted codes (the 8 previous codes, the current
code, and the 8 next codes). This will permit for a time skew of up to 4 minutes
between client and server.
Do you want to do so? (y/n) y

If the computer that you are logging into isn't hardened against brute-force
login attempts, you can enable rate-limiting for the authentication module.
By default, this limits attackers to no more than 3 login attempts every 30s.
Do you want to enable rate-limiting? (y/n) y

┌──(kali㉿kali)-[~]
└─$ 
```

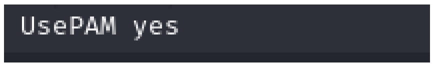
CONFIGURE SSH DAEMON FOR 2FA

- Open SSH server configuration file.

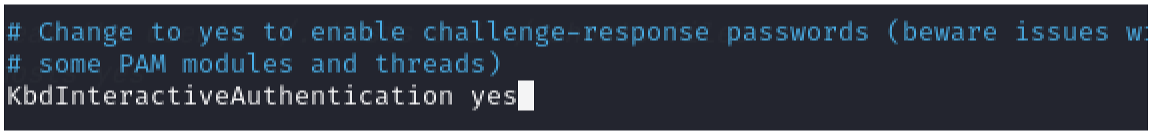


```
┌──(kali㉿kali)-[~]
└─$ sudo nano /etc/ssh/sshd_config
```

- Ensure UsePAM and KbdInterativeAuthenticatiion parameters are set to 'yes'.

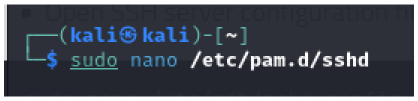


```
UsePAM yes
```



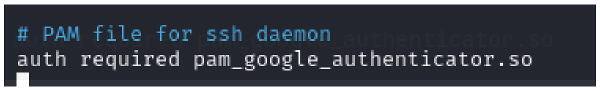
```
# Change to yes to enable challenge-response passwords (beware issues wi
# some PAM modules and threads)
KbdInteractiveAuthentication yes
```

- Edit the PAM rule file for SSH daemon.



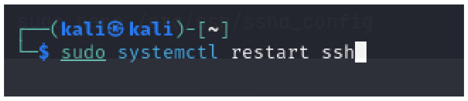
```
┌──(kali㉿kali)-[~]
└─$ sudo nano /etc/pam.d/sshd
```

- Add the line:



```
# PAM file for ssh daemon
auth required pam_google_authenticator.so
```

- Restart SSH daemon.



```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl restart ssh
```

CONCLUSION

In conclusion, the implementation of Multi-Factor Authentication (MFA) has significantly enhanced the security of user accounts and sensitive data. By utilizing a combination of factors such as passwords and Google Authenticator, we have strengthened the defense against unauthorized access and potential cyber threats. The process of configuring MFA, particularly through the integration of SSH with Google Authenticator, has been streamlined and effective. Although MFA requires additional steps for users during login, the added security far outweighs the inconvenience. As cybersecurity threats continue to evolve, adopting robust solutions like MFA is essential for safeguarding both organizational and personal assets. This initiative not only aligns with industry standards but also supports the organization’s broader goal of fostering a secure, digital environment for all stakeholders.