Topic:

MFA

Task:

“Set up MFA for your machine for securely accessing it remotely from another machine”.

Task Assign by: **Digital Empowerment Networks.**

Done by: **Khawar Amin**

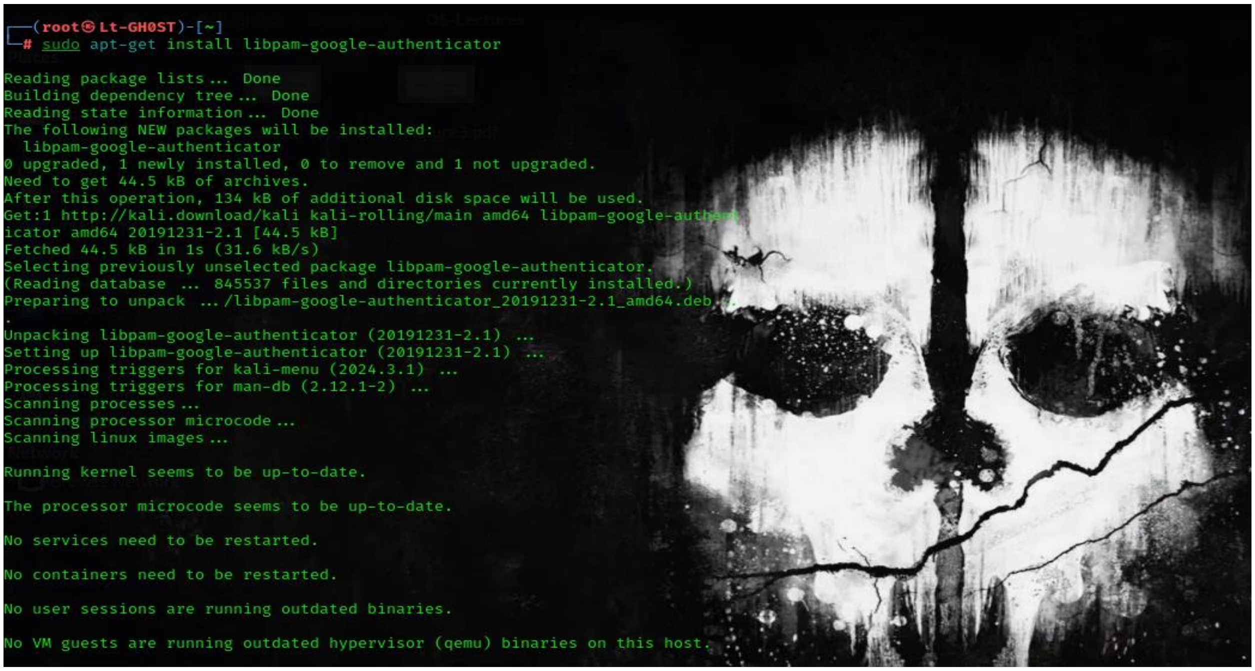
Step 1:

My OS: Kali Linux.

Username: lt-gh0st.

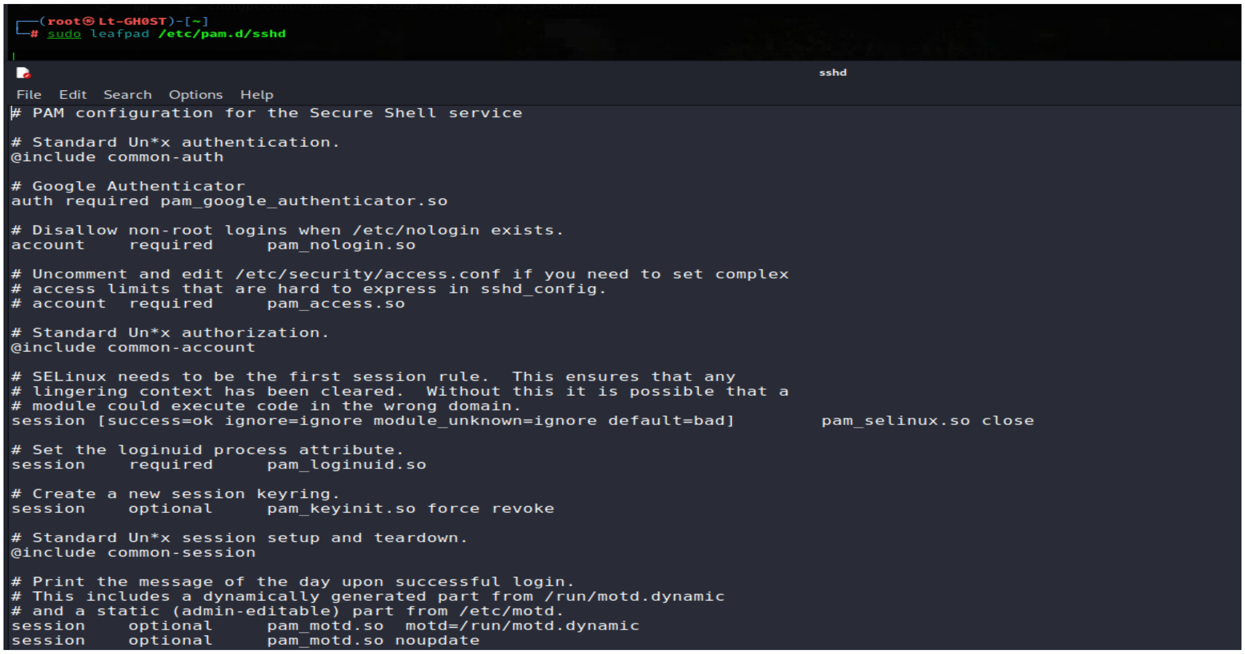
Other device OS: Windows.

We need to Download MFA for us. We can use different vendors like Google and Microsoft. I will be using Google MFA for this task. Let's download it using the following commands.



```
┌──(root㉿Lt-GH0ST)-[~]
└─# sudo apt-get install libpam-google-authenticator

Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  libpam-google-authenticator
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 44.5 kB of archives.
After this operation, 134 kB of additional disk space will be used.
Get:1 http://kali.download/kali kali-rolling/main amd64 libpam-google-authent
icator amd64 20191231-2.1 [44.5 kB]
Fetched 44.5 kB in 1s (31.6 kB/s)
Selecting previously unselected package libpam-google-authenticator.
(Reading database ... 845537 files and directories currently installed.)
Preparing to unpack .../libpam-google-authenticator_20191231-2.1_amd64.deb ..
.
Unpacking libpam-google-authenticator (20191231-2.1) ...
Setting up libpam-google-authenticator (20191231-2.1) ...
Processing triggers for kali-menu (2024.3.1) ...
Processing triggers for man-db (2.12.1-2) ...
Scanning processes...
Scanning processor microcode...
Scanning linux images...

Running kernel seems to be up-to-date.

The processor microcode seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

Now Open the SSH PAM configuration file as shown in following screenshot



```
┌──(root㉿Lt-GH0ST)-[~]
└─# sudo leafpad /etc/pam.d/sshd
```

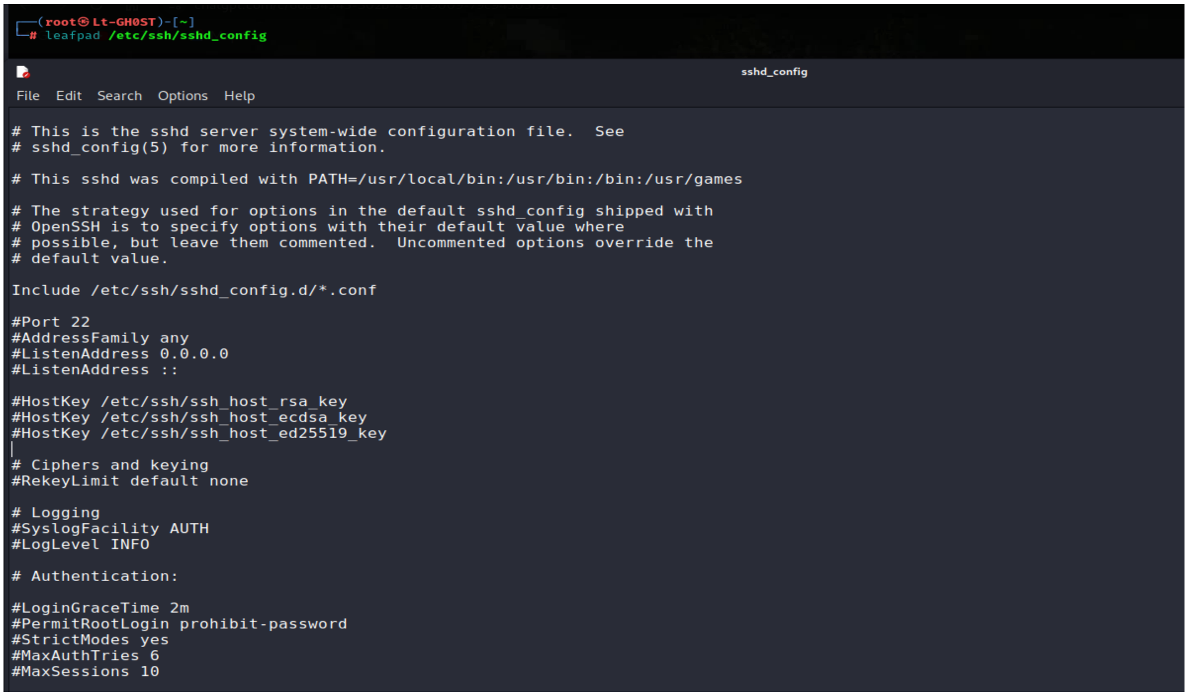
```
# PAM configuration for the Secure Shell service

# Standard Un*x authentication.
@include common-auth

# Google Authenticator
auth required pam_google_authenticator.so

# Disallow non-root logins when /etc/nologin exists.
account    required     pam_nologin.so

# Uncomment and edit /etc/security/access.conf if you need to set complex
# access limits that are hard to express in sshd_config.
# account  required     pam_access.so

# Standard Un*x authorization.
@include common-account

# SELinux needs to be the first session rule.  This ensures that any
# lingering context has been cleared.  Without this it is possible that a
# module could execute code in the wrong domain.
session [success=ok ignore=ignore module_unknown=ignore default=bad]        pam_selinux.so close

# Set the loginuid process attribute.
session    required     pam_loginuid.so

# Create a new session keyring.
session    optional     pam_keyinit.so force revoke

# Standard Un*x session setup and teardown.
@include common-session

# Print the message of the day upon successful login.
# This includes a dynamically generated part from /run/motd.dynamic
# and a static (admin-editable) part from /etc/motd.
session    optional     pam_motd.so  motd=/run/motd.dynamic
session    optional     pam_motd.so noupdate
```

Cmd: leafpad /etc/pam.d/sshd.

You can use any editor like Nano, Vim or leafpad I am using leafpad and add that line in it “**auth required pam_google_authenticator.so**”.

Now it's time to edit SSH daemon configuration file. as shown in screenshot.

Cmd: leafpad /etc/ssh/sshd_config.



```
┌──(root㉿Lt-GH0ST)-[~]
└─# leafpad /etc/ssh/sshd_config
```

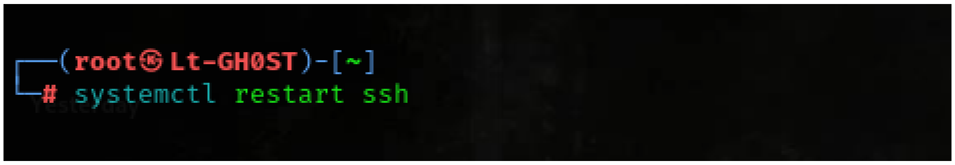
```
# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

And edits these lines to yes.

ChallengeResponseAuthentication yes.

UsePAM yes.

Now restart the SSH service in kali.

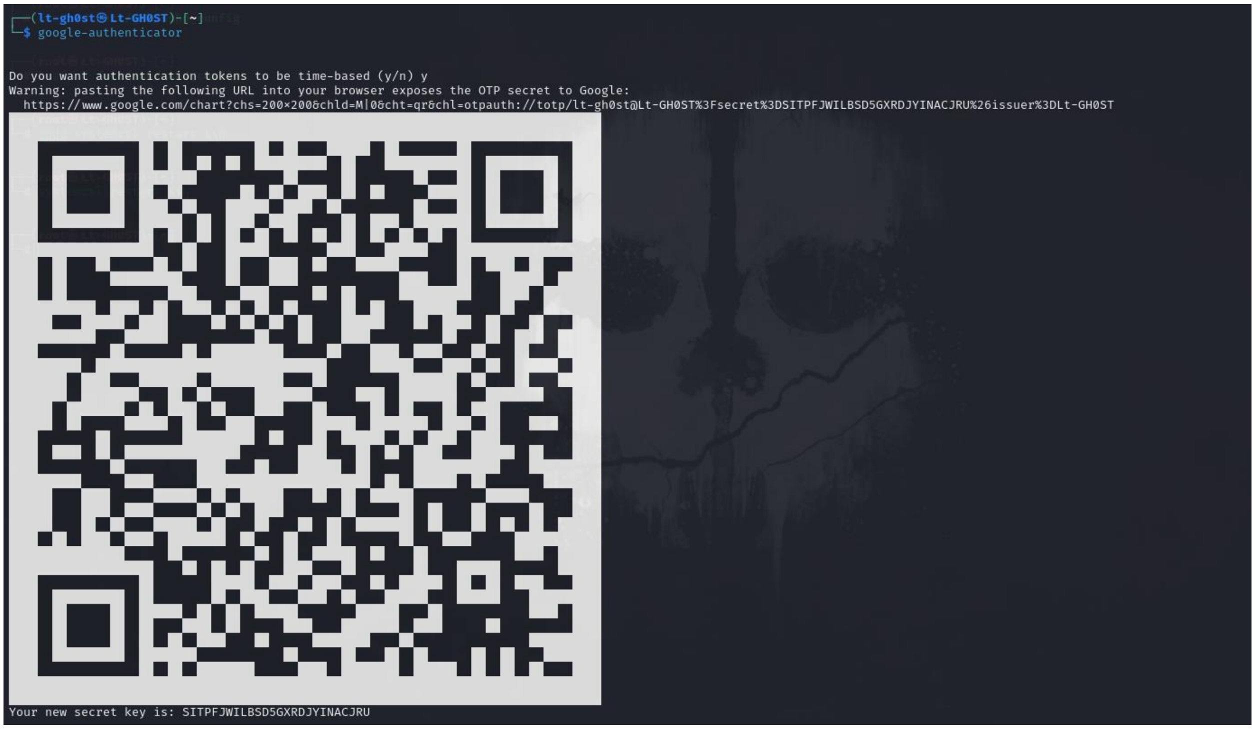


```
┌──(root㉿Lt-GH0ST)-[~]
└─# systemctl restart ssh
```

Now run the following command to set up Google MFA.

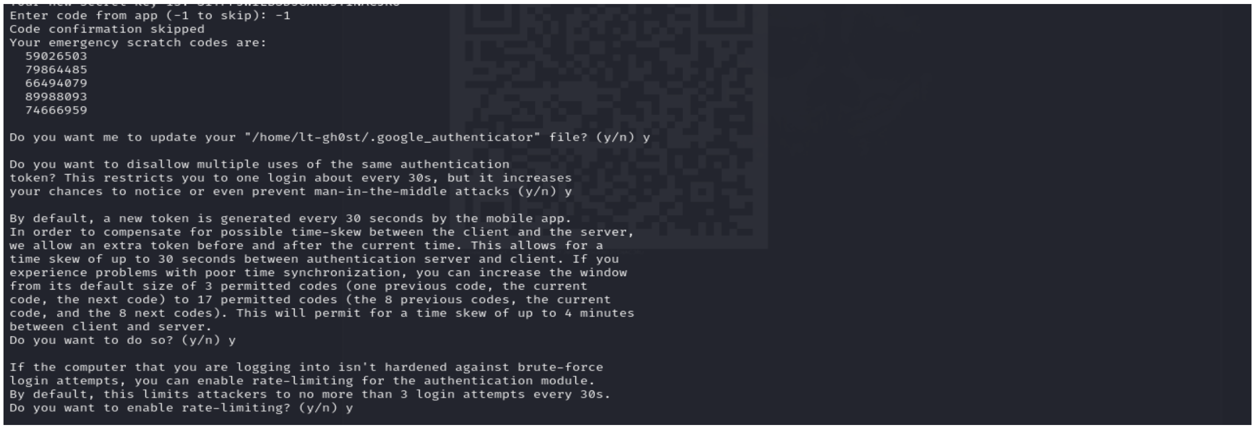
```
┌──(lt-gh0st㉿Lt-GH0ST)-[~]
```

└─$google-authenticator



```
┌──(lt-gh0st㉿Lt-GH0ST)-[~]
└─$ google-authenticator

Do you want authentication tokens to be time-based (y/n) y
Warning: pasting the following URL into your browser exposes the OTP secret to Google:
  https://www.google.com/chart?chs=200x200&chld=M|0&cht=qr&chl=otpauth://totp/lt-gh0st@Lt-GH0ST%3Fsecret%3DSITPFJWILBSD5GXRDJYINACJRU%26issuer%3DLt-GH0ST
```

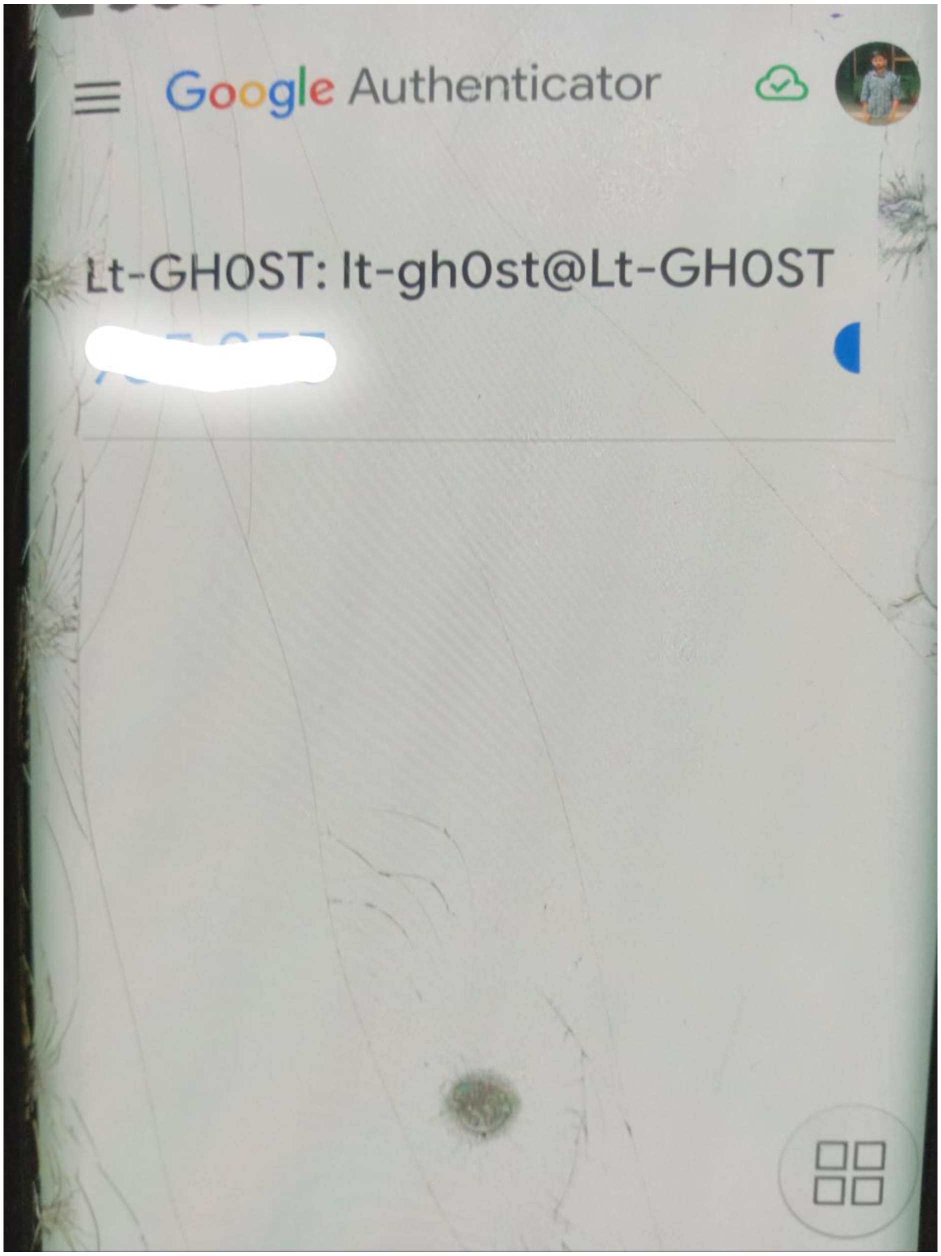
```
Your new secret key is: SITPFJWILBSD5GXRDJYINACJRU
```



```
Enter code from app (-1 to skip): -1
Code confirmation skipped
Your emergency scratch codes are:
  59026503
  79864485
  66494079
  89988093
  74666959

Do you want me to update your "/home/lt-gh0st/.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) y

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
```

I used Timed based token for MFA. and don't forget to save the configuration by pressing y and codes.

And Scan this QR code using Google MFA App. To get the verification codes.



Google Authenticator

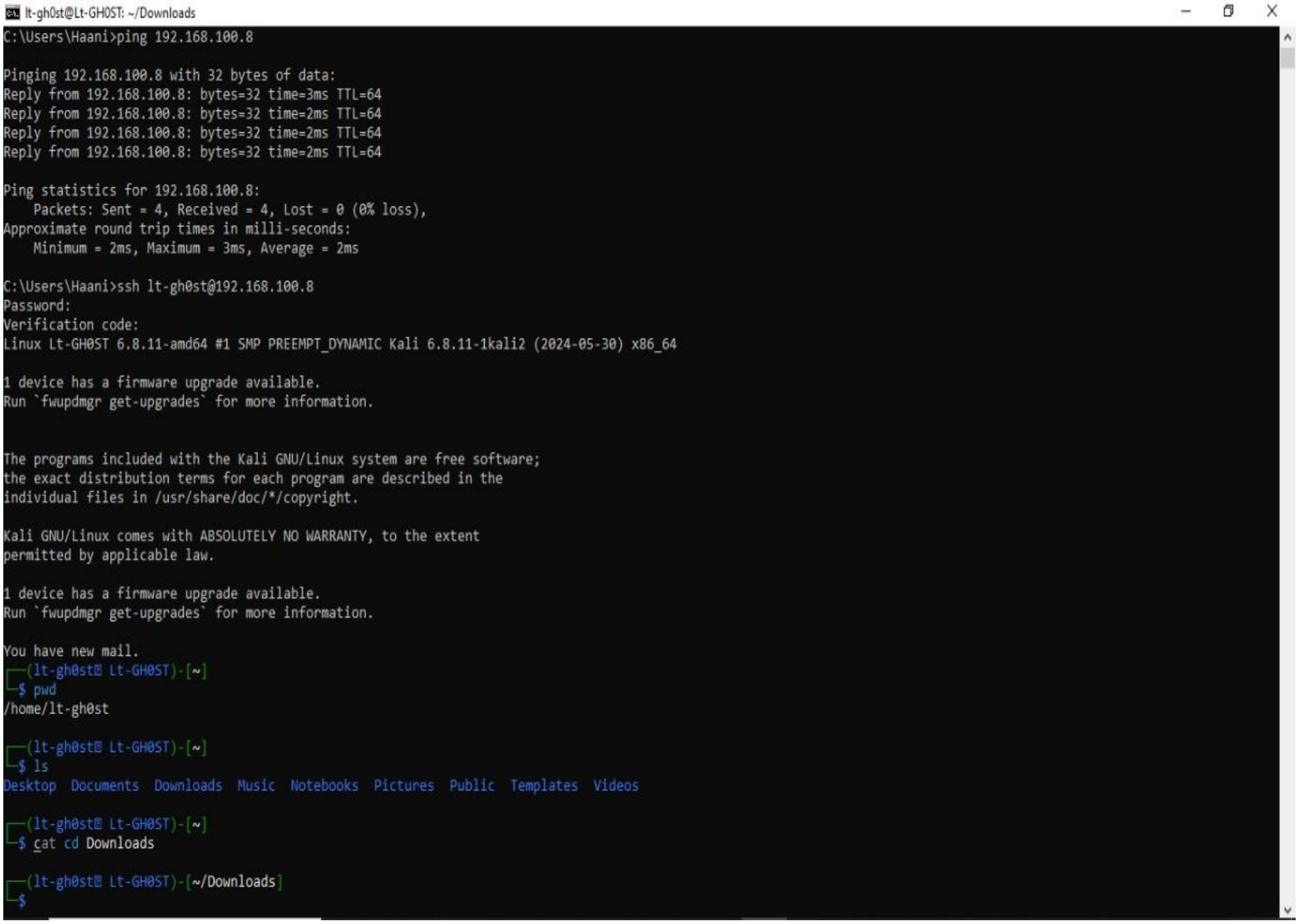
Lt-GHOST: lt-gh0st@Lt-GHOST

[redacted]

Now it's time to download putty on windows machine to remotely access my Linux machine or simply use the cmd of windows to connect to Linux device. I am using cmd as it is very easy. Below screenshot shows every step.

Command: ssh username@ip_address

After that enter password and verification code from app



```
C:\Users\Haani>ping 192.168.100.8

Pinging 192.168.100.8 with 32 bytes of data:
Reply from 192.168.100.8: bytes=32 time=3ms TTL=64
Reply from 192.168.100.8: bytes=32 time=2ms TTL=64
Reply from 192.168.100.8: bytes=32 time=2ms TTL=64
Reply from 192.168.100.8: bytes=32 time=2ms TTL=64

Ping statistics for 192.168.100.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 3ms, Average = 2ms

C:\Users\Haani>ssh lt-gh0st@192.168.100.8
Password:
Verification code:
Linux Lt-GH0ST 6.8.11-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.8.11-1kali2 (2024-05-30) x86_64

1 device has a firmware upgrade available.
Run `fwupdmgr get-upgrades` for more information.


The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

1 device has a firmware upgrade available.
Run `fwupdmgr get-upgrades` for more information.

You have new mail.
┌──(lt-gh0st□ Lt-GH0ST)-[~]
└─$ pwd
/home/lt-gh0st

┌──(lt-gh0st□ Lt-GH0ST)-[~]
└─$ ls
Desktop  Documents  Downloads  Music  Notebooks  Pictures  Public  Templates  Videos

┌──(lt-gh0st□ Lt-GH0ST)-[~]
└─$ cat cd Downloads

┌──(lt-gh0st□ Lt-GH0ST)-[~/Downloads]
└─$
```

First cmd show I am in C drive of Haani then after remotely access procedure using MFA i am able to access my linux machines you can clearly see in screenshot. Thank you.