First of all we need to change the installation shell script for the postgres database.

We need to change the following line of pg\_hba.conf file from

local all all md5

to

local all all trust

This will allow the shell script to be postgres db user to run psql command to create pegarole or postgres database and change the password of those users, create pegadb database and grant role. If the local md5 is there then it will ask for password of postgres nd script will not move forward.

So using the trust shell script will be able to create databae and users and all that and script will be successful.

Also this file has ssl authentication methods md5 and also non-ssl authentication method to md5 , which means during the installation it will allow non-ssl and even after getting ssl certs and install ssl on database it will also allow ssl-host , may be we can disable non-ssl after installing the ssl certs for db for better security.

After creating database we need to change back this line from trust to md5 again because of Nessus scan will raise this issue and this is a security findings.

local all all trust

host all all 127.0.0.1/32 md5

host all all 10.45.76.0/24 md5

hostssl all all 127.0.0.1/32 md5 clientcert=0

hostssl all all 10.45.76.0/24 md5 clientcert=0

This pg\_hba.conf file is hardened already and collected from existing pega database after applying all security fixes of Nessus scan and STIGs.

Similarly I had to change the postgres.conf file

The following line means postgres will allow remote connection request , so that application server postgres or remote pgadmin server can access to this db server from remote.

listen\_addresses = '\*'

Following line means the database is good with 100 concurrent user sessions, so for production we need to make it 200-500 , if we want to support 500 concurrent users for pega.

max\_connections = 100 # (change requires restart)

then disable ssl because during the installation ssl certs are not installed,

##ssl = on

ssl = off

##ssl\_ca\_file = '/var/lib/pgsql/11/data/ssl/pega88-database-node.sil.arl.psu.edu.chain.cer'

##ssl\_cert\_file = '/var/lib/pgsql/11/data/ssl/pega88-database-node.sil.arl.psu.edu.crt'

#ssl\_crl\_file = ''

##ssl\_key\_file = '/var/lib/pgsql/11/data/ssl/pega88-database-node.sil.arl.psu.edu.key'

#ssl\_ciphers = 'HIGH:MEDIUM:+3DES:!aNULL' # allowed SSL ciphers

ssl\_ciphers = 'TLSv1.2+HIGH:!aNULL' # allowed SSL ciphers

ssl\_prefer\_server\_ciphers = on

after the pega db installation and ssl certs installation I had to revert back to ===

# - SSL -

ssl = on

#ssl = off

ssl\_ca\_file = '/var/lib/pgsql/11/data/ssl/ey-postgresql.sil.arl.psu.edu.chain.cer'

ssl\_cert\_file = '/var/lib/pgsql/11/data/ssl/ey-postgresql.sil.arl.psu.edu.cer'

#ssl\_crl\_file = ''

ssl\_key\_file = '/var/lib/pgsql/11/data/ssl/ey-postgresql.sil.arl.psu.edu.key'

#ssl\_ciphers = 'HIGH:MEDIUM:+3DES:!aNULL' # allowed SSL ciphers

ssl\_ciphers = 'TLSv1.2+HIGH:!aNULL' # allowed SSL ciphers

ssl\_prefer\_server\_ciphers = on

#ssl\_ecdh\_curve = 'prime256v1'

#ssl\_dh\_params\_file = ''

#ssl\_passphrase\_command = ''

#ssl\_passphrase\_command\_supports\_reload = off

Also no need to make any change of following part:

#------------------------------------------------------------------------------

# CUSTOMIZED OPTIONS

#------------------------------------------------------------------------------

# Add settings for extensions here

pljava.classpath = '/usr/pgsql-11/lib64/pljava.jar'

pljava.vmoptions = '-Xms32M -Xmx64M -XX:ParallelGCThreads=2'

work\_mem = 5MB

pljava.libjvm\_location = '/usr/java/jdk-11/lib/server/libjvm.so'

==🡺 after making change on those 2 configuration files we need to check the installation shell script

And run in the postgres database vm as root.

# ./pega\_installation\_shell\_script.sh

This shell script will install latest version of oracle jdk , so before executing this script check the latest version of jdk and download that and use that version in this script.

It willalso install latest version of postgres-11 , so download and change script accordingly.

It is a critical finding in Nessus scan to use latest version of postgres and jdk version , otherwise we have to upgrade both of them aster installation.

After installation try to check as below

# su – postgres

$ psql

\l

Will list all databae

\q

It will not show TLS

Or connect to this database from pgadmin which is installed in windows jump server.

After installation request for SSL certs for postgres database server.

Once ssl cert is available then install on the server.

Copy 3 files to SSL folder

Change permission to 600 to key file and change to 640 on other 2 files , also make sure the owner is postgres:root of those 3 files and then put the name of those certs with location to postgres.conf file and turn on ssl = on and restart postgres db by

# systemctl restart postgres-11.service

DONE