Using active directory to authenticatication

What is an active directory?

is a directory service developed by Microsoft for Windows domain networks. It is included in most Windows Server operating systems as a set of processes and services.

Why active directory use?

To centralized domain management.

How?

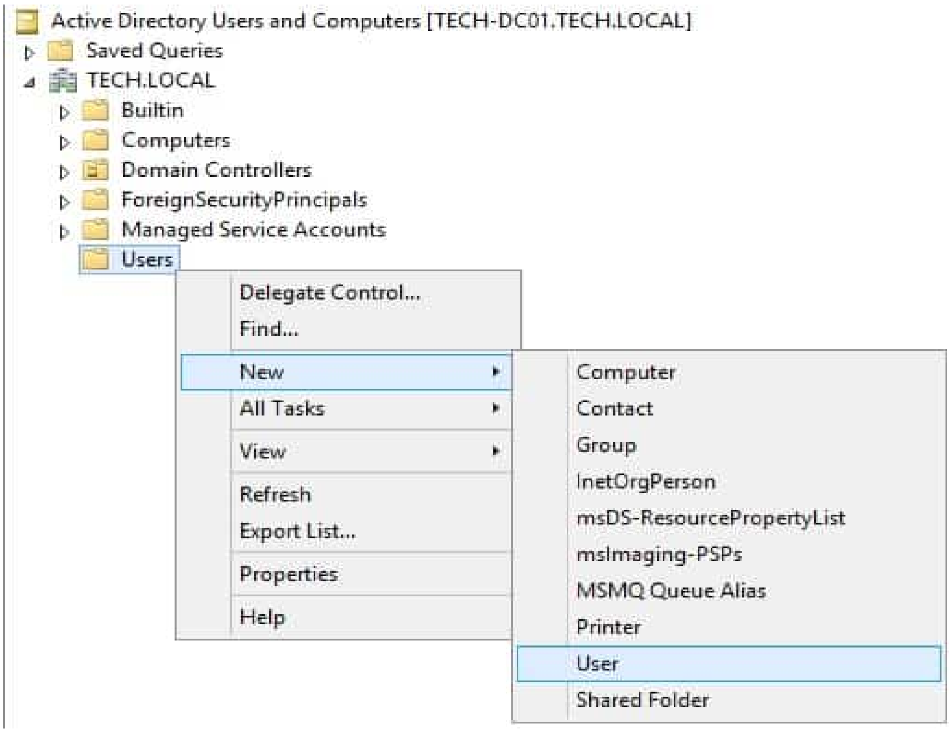
Instead of signing in using django authentication, we want the authentication process tobe done using a windows server that has an active directory service.

Requirements:

1- Windows server that has an active directory service.
2- Django applications.

Steps:

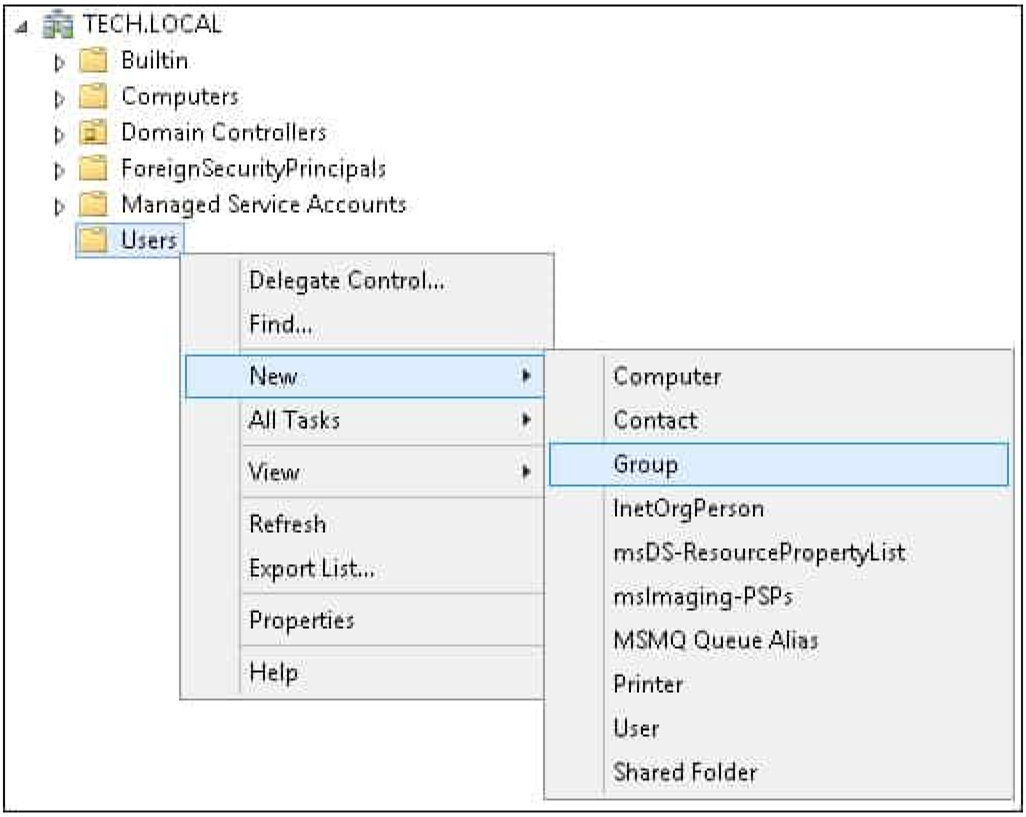
1- create an account in our active directory.
userName='admin'
password='Admin@123'



This account will be used to authenticate as admin on the Django web interface.

2- By the same way, create another account to query the passwords stored on the Active Directory database.
UserName = 'bind'
Password = 'Bind@123'

3- create a group on the Active directory database: group name =’django-admins’



4- Add the admin user as a member of the django-admins group.

5- Install LDAP:
-apt-get -y install libldap2-dev libsasl2-dev ldap-utils
-pip3 install django-auth-ldap

6- setup your configurations in setting.py file:
First add the code below in setting.py:

```
import ldap
from django_auth_ldap.config import LDAPSearch

AUTH_LDAP_SERVER_URI = 'ldap://<IP: FOR AD>’
AUTH_LDAP_BIND_DN = "CN=bind,CN=Users,DC=tech,DC=local"
AUTH_LDAP_BIND_PASSWORD = "kamisama123@"
AUTH_LDAP_USER_SEARCH = LDAPSearch(
        "dc=tech,dc=local", ldap.SCOPE_SUBTREE, "sAMAccountName=%(user)s"
        )

AUTH_LDAP_USER_ATTR_MAP = {
        "username": "sAMAccountName",
            "first_name": "givenName",
                "last_name": "sn",
                    "email": "mail",
}
from django_auth_ldap.config import ActiveDirectoryGroupType
AUTH_LDAP_GROUP_SEARCH = LDAPSearch(
        "dc=tech,dc=local", ldap.SCOPE_SUBTREE, "(objectCategory=Group)"
        )
AUTH_LDAP_GROUP_TYPE = ActiveDirectoryGroupType(name_attr="cn")
AUTH_LDAP_USER_FLAGS_BY_GROUP = {
        "is_superuser": "CN=django-admins,CN=Users,DC=TECH,DC=LOCAL",
        "is_staff": "CN=django-admins,CN=Users,DC=TECH,DC=LOCAL",
        }
AUTH_LDAP_FIND_GROUP_PERMS = True
AUTH_LDAP_CACHE_GROUPS = True
AUTH_LDAP_GROUP_CACHE_TIMEOUT = 1  # 1 hour cache

AUTHENTICATION_BACKENDS = [
        'django_auth_ldap.backend.LDAPBackend',
          'django.contrib.auth.backends.ModelBackend',
]
```

Then change the following variables:

-Domain controller IP - 192.168.15.10:
AUTH_LDAP_SERVER_URI = 'ldap://<IP: FOR AD>' : replace <IP: FOR AD> with the ip for the active directory. For example: 'ldap://192.168.41.146'

-Active directory domain - dc=tech,dc=local:
You can specify a DN to use as the search base, for example ldap://melbourne.example.com/dc=zain,dc=com. This specifies the part of the LDAP directory used to search for the user identity.

-Bind user - CN=bind,CN=Users,DC=tech,DC=local:
A user profile in the same organizational unit as the user's LDAP object but with the name cn=LDAP Profile.
For example, dc=com/dc=zain/cn=Sales(group)/cn=LDAP Profil(username).
Bind user password - kamisama123@:
Replace it with ‘Admin@123’

- Group permission - Members of the django-admin group will have total access to the web interface

*Keep in mind that you need to change this to reflect your network environment.

7- run your django project, On the login screen, use the Django username and password created before.

username: ‘admin’

password Enter the Active directory password(‘Admin@123’)