LDAP

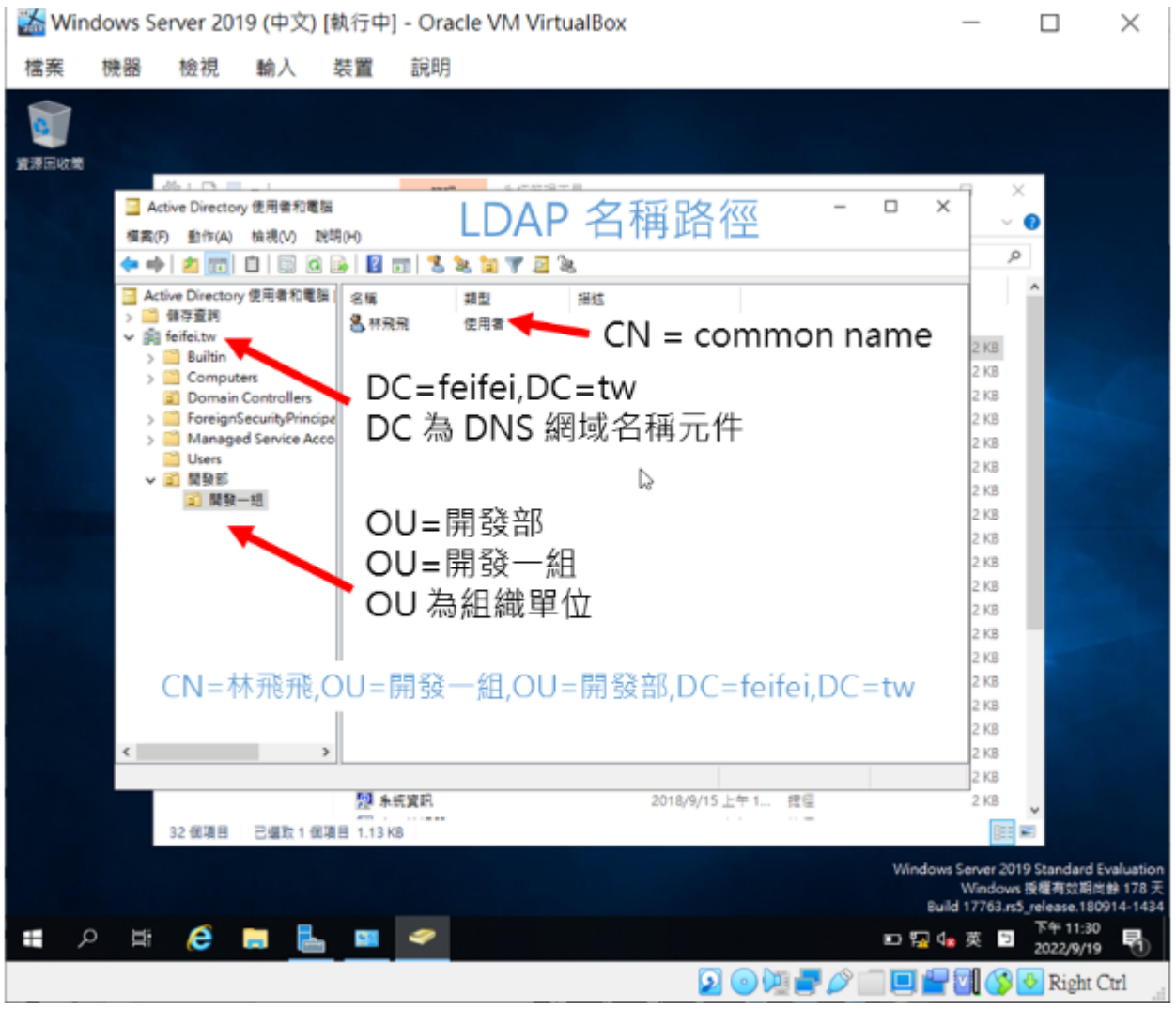
Created	@2024年8月2日 下午1:55
Tags	

LDAP 為一個輕型目錄協定，可以讀取電腦使用者帳號並驗證是否有此帳密

詳細 LDAP 說明可以查看此文章

1. https://ithelp.ithome.com.tw/articles/10298554
2. https://blog.poychang.net/ldap-introduction/

LDAP UI介面



這邊要注意的是要先懂 LDAP 的查詢概念

名詞	說明	範例
DN	描述物件路徑	1. CN=總務採購,OU=業務管理室,OU=UITC,DC=uitctech,DC=com,DC=tw 2. 實際畫面查找為 uitctech.com.tw ⇒ UITC ⇒ 業務管理室 ⇒ CN 3. 順序不能變
DC	相當於一個網域	1. DC=uitctech,DC=com,DC=tw ⇒ 在畫面上相當於 uitctech.com.tw
CN	使用者資訊	1. CN=總務採購
OU	組織架構	1. OU=UITC

目前官網有自己套件但文件跟實作偏少，所以使用此套件

`Novell.Directory.Ldap.NETStandard`

而且此套件在自己 GitHub有寫許多範例可以參考

參考實作

1. https://blog.poychang.net/use-thired-party-package-to-implement-ldap-authenticate-in-dotnet-core/
2. https://dotblogs.com.tw/OldNick/2023/09/13/DOTNET_LDAP_LDAPS
3. https://rainmakerho.github.io/2019/09/17/2019025/
4. https://github.com/dsbenghe/Novell.Directory.Ldap.NETStandard/blob/master/original_samples/Samples/Search.cs

目前實作如下

appSetting Options

```
public class LDAPOptions
{
    /// <summary>
    /// Ldap Server 192.168.10.250
    /// </summary>
    [Required]
    public string LdapServer { get; set; }
    /// <summary>
    /// 預設 389
    /// </summary>
    [Required]
    public int Port { get; set; }
    /// <summary>
    /// Admin 帳號 => 用來搜尋
    /// </summary>
    [Required]
    public string AdminUser { get; set; }
    [Required]
    public string AdminMima { get; set; }
    /// <summary>
    /// OU=UITC,DC=uitctech,DC=com,DC=tw
    /// </summary>
    [Required]
    public string BaseDN { get; set; }
    /// <summary>
    /// uitctech.com.tw
    /// </summary>
    [Required]
    public string Domain { get; set; }
}
```

返回資訊

```
public class LDAPInfo
{
    /// <summary>
    /// 範例: 陳曉明
    /// 一定要有值
    /// </summary>
    public string? DisplayName { get; set; }
    /// <summary>
    /// memberOf = CN=資訊服務部,OU=資訊服務部,OU=UITC,DC=uitctech,DC=com,DC=tw
    /// 通常這個要大於0，不然他可能是電腦帳號而已
    /// </summary>
    public List<string> MemberOf { get; set; } = new();
```

```
    /// <summary>
    /// chenming
    /// </summary>
    public string? SAMAccountName { get; set; }
    /// <summary>
    /// chenming@uitc.com.tw
    /// </summary>
    public string? UserPrincipalName { get; set; }
}
```

1. 取得 Connection

LdapServer	172.28.33.60
Port	預設389
user	帳號
password	密碼

```
    private LdapConnection GetConnection(string user, string password)
    {
        LdapConnection connection = new LdapConnection();
        connection.Connect(_ldapOptions.LdapServer, _ldapOptions.Port);
        connection.Bind(user, password);
        return connection;
    }
```

2. 驗證是否有此帳號

關鍵在於 connection.Bound 會回傳 true / false

```
    public bool VaildLDAPAuth(string username, string password)
    {
        try
        {
            string user = $"{username}@{_ldapOptions.Domain}";
            using (var connection = GetConnection(user, password))
            {
                return connection.Bound;
            }
        }
        catch (Exception ex)
        {
            _logger.LogError("{@Username}-LDAP驗證失敗：Error：{@Error}", username,ex);
            return false;
        }
    }
```

3. 查詢所有使用者

connection.Search(_ldapOptions.BaseDN, LdapConnection.ScopeSub, searchFilter, attrList, false) 參數說明

1. 預設基本絕對位置 ⇒ 如 OU=UITC,DC=uitctech,DC=com,DC=tw
2. 查找資料位置 ⇒ LdapConnection.ScopeSub 查詢目前 DN 下的樹（通常會使用此設定）
3. searchFilter 需要過濾的條件 ⇒ 如 (objectClass=user) 我只找到 objectClass 為 user，
   也可以鎖定某些屬性如 sAMAccountName 只要此帳號 就會變成(&(objectClass=user)(sAMAccountName={0}))

4. attrList 需要回傳的欄位常用為

memberOf	組織架構(可以多個)	CN=資訊服務部,OU=資訊服務部,OU=UITC,DC=uitctech,DC=com,DC=tw
displayName	姓名	莊理峻
sAMAccountName	帳號	lijungjhuang
userPrincipalName	完整帳號	lijungjhuang@uitctech.com.tw

5. typesOnly true 只返回名稱，false 返回名稱及屬性

上述參數參考 https://blog.poychang.net/use-thired-party-package-to-implement-ldap-authenticate-in-dotnet-core/

非常詳細

```
public List<LDAPInfo> SearchUsersAll()
{
    try
    {
        string user = _uitcSecurityHelper.DecryptData(_ldapOptions.AdminUser);
        string mima = _uitcSecurityHelper.DecryptData(_ldapOptions.AdminMima);

        using (var connection = GetConnection(user, mima))
        {
            string searchFilter = "(objectClass=user)";
            string[] attrList = new string[] { 組織架構, 姓名, 帳號, 完整帳號 };
            var lsc = connection.Search(_ldapOptions.BaseDN, LdapConnection.ScopeSub, searcl

            List<LDAPInfo> result = new List<LDAPInfo>();
            while (lsc.HasMore())
            {
                LdapEntry nextEntry = lsc.Next();

                if (nextEntry is null)
                {
                    continue;
                }

                LDAPInfo dto = new LDAPInfo();
                foreach (var item in nextEntry.GetAttributeSet())
                {
                    if (item.Name == 姓名)
                    {
                        dto.DisplayName = item.StringValue;
                    }
                    else if (item.Name == 組織架構)
                    {
                        dto.MemberOf.AddRange(item.StringValueArray);
                    }
                    else if (item.Name == 帳號)
                    {
                        dto.SAMAccountName = item.StringValue;
                    }
                    else if (item.Name == 完整帳號)
                    {
                        dto.UserPrincipalName = item.StringValue;
                    }
                }
```

```
                    result.Add(dto);
                }
                return result;
            }
        }
        catch (Exception ex)
        {
            _logger.LogError("LDAP取得所有ADUser失敗，Appsetting：{@Appsetting}，Error：{@Error}",
            return new List<LDAPInfo>();
        }
    }
```

4. 查詢單筆使用者

```
    public LDAPInfo? SearchBySAMAccountName(string samAccountName)
    {
        try
        {
            string user = _uitcSecurityHelper.DecryptData(_ldapOptions.AdminUser);
            string mima = _uitcSecurityHelper.DecryptData(_ldapOptions.AdminMima);

            using (var connection = GetConnection(user, mima))
            {
                string searchFilter = "(&(objectClass=user)(sAMAccountName={0}))";
                string[] attrList = new string[] { 組織架構, 姓名, 帳號,完整帳號 };
                var lsc = connection.Search
                            (
                                _ldapOptions.BaseDN,
                                LdapConnection.ScopeSub,
                                String.Format(searchFilter, samAccountName),
                                attrList,
                                false
                            );

                if (!lsc.HasMore())
                {
                    return null;
                }

                LDAPInfo info = new LDAPInfo();
                while (lsc.HasMore())
                {
                    LdapEntry nextEntry = lsc.Next();
                    var account = nextEntry.GetAttribute(帳號);
                    if (account != null && account.StringValue == samAccountName)
                    {
                        foreach (var item in nextEntry.GetAttributeSet())
                        {
                            if (item.Name == 姓名)
                            {
                                info.DisplayName = item.StringValue;
                            }
                            else if (item.Name == 組織架構)
                            {
                                info.MemberOf.AddRange(item.StringValueArray);
                            }
```

```
                        else if (item.Name == 帳號)
                        {
                            info.SAMAccountName = item.StringValue;
                        }
                        else if (item.Name == 完整帳號)
                        {
                            info.UserPrincipalName = item.StringValue;
                        }
                    }
                    break;
                }
            }
            return info;
        }

    }
    catch (Exception ex)
    {
        _logger.LogError("LDAP取得單筆ADUser失敗，samAccountName：{@SamAccountName}e，Appsettin
        return null;
    }
}
```