Лабораторна робота №10

Основи використання PowerShell для адміністрування домену Active Directory

Мета: навчитися основних прийомів застосування командної оболонки PowerShell для роботи з Active Directory.

Інструменти: гіпервізор VirtualBox; (за потреби) мережний емулятор GNS3; модель комп'ютерної мережі та структура домену, побудовані у межах лабораторних робіт №1-9.

Теоретичні відомості та завдання до лабораторної роботи

ОКРЕМІ КОМАНДИ POWERSHELL ДЛЯ РОБОТИ З ACTIVE DIRECTORY

Командна оболонка PowerShell у Windows Server доступна з головною меню. Звідси можна викликати і саму оболонку PowerShell, і середовище для написання скриптів PowerShell ISE (рис. 10.1). Починати легше з Windows PowerShell, а скрипти доцільно писати і перевіряти у Windows PowerShell ISE.

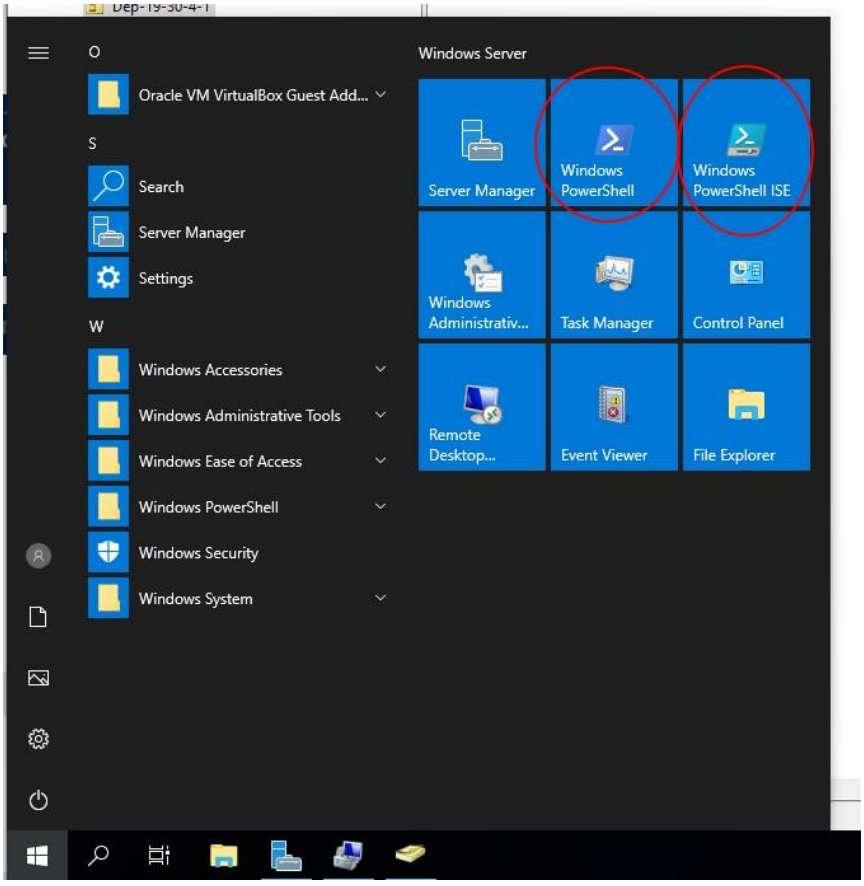


Рис. 10.1

Одержання відомостей про домен

Загальні відомості про домен виводить командлет Get-ADDomain. Введений без параметрів, цей командлет дає вивід, подібний до наступного:

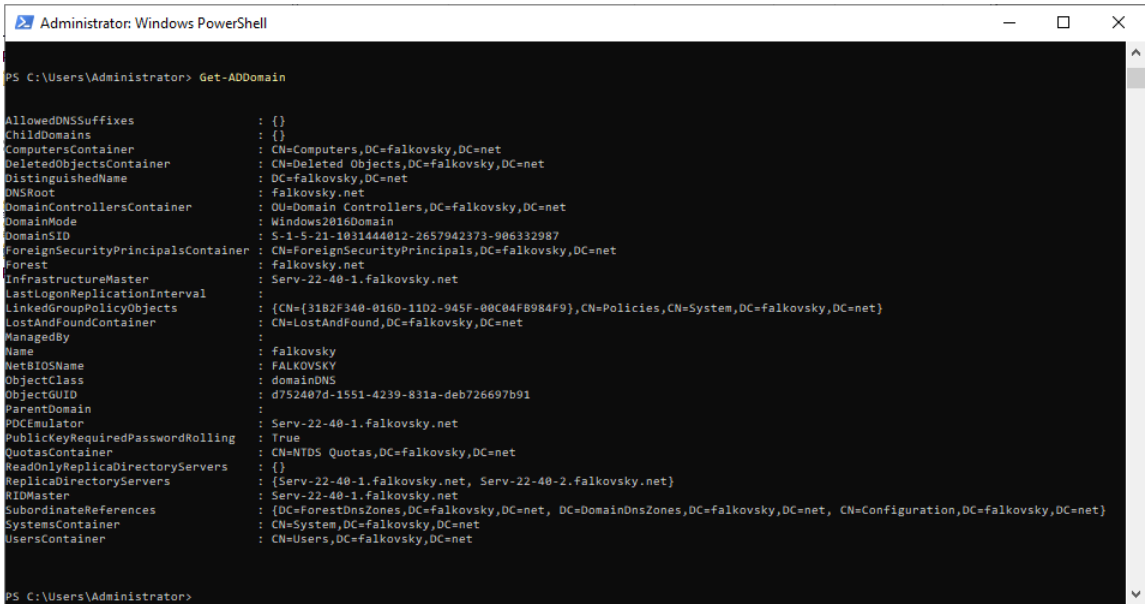


Рис. 10.2

Зверніть увагу на формат імен. Кожна складова доменного імені вказується окремо і супроводжується відомостями про структурний рівень.

Наприклад, поле DomainControllerContainer має вигляд:

```
DomainControllersContainer          : OU=Domain Controllers,DC=falkovsky,DC=net
```

Тобто *Domain Controllers* міститься на рівні організаційних одиниць (OU), що належить до домену *holovnia.net* (DC).

А ось поле DomainControllerContainer:

```
ComputersContainer                  : CN=Computers,DC=falkovsky,DC=net
```

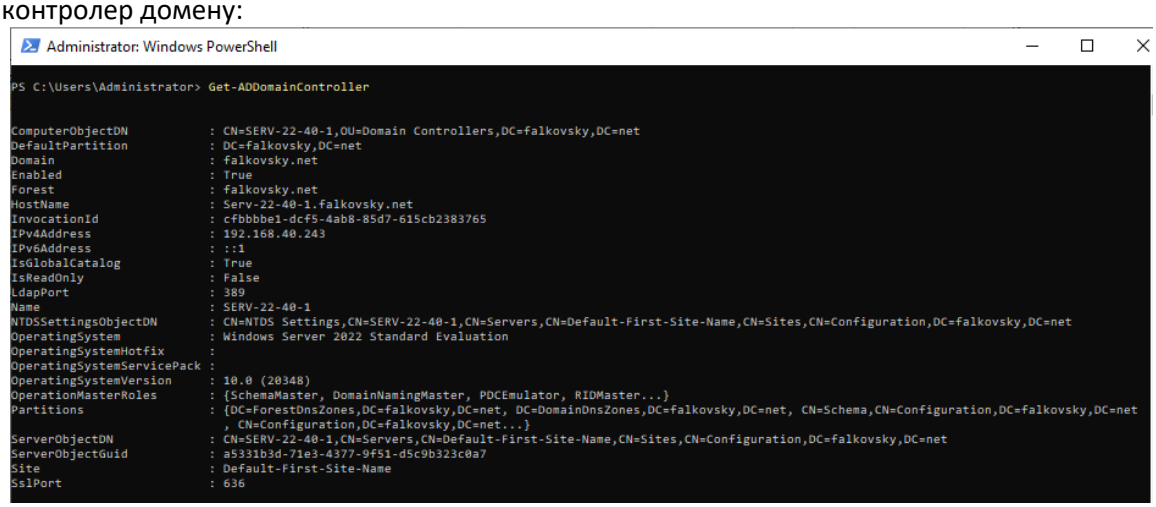
Тут *Computers* позначається як CN (*Common Name*), що у випадку Active Directory відповідає за контейнер.

Пояснення до деяких полів

- ChildDomains - дочірні домени (якщо є)
- ComputersContainer - контейнер для комп’ютерів домену, які не є контролерами домену (наприклад, робочих станцій)
- DeletedObjectsContainer - контейнер для видалених об’єктів
- DestinguishedName - унікальне ім’я домену
- DNSRoot - домен верхнього рівня
- DomainControllerContainer - контейнер для контролерів домену
- DomainSID - SID домену (зазвичай утворюється на основі SID першого контролеру домену)
- Forest - ім’я лісу (у невеликих мережах на основі AD - часто тотожне імені домену)
- LinkedGroupPolicyObjects - об’єкти групової політики рівня домену
- Name - скорочене ім’я домену (те, яке використовується під час входу користувачів, наприклад)

- ParentDomain - ім’я батьківського домену (якщо є)
- ReplicaDirectoryServers - сервери, які беруть участь у реплікації
- SystemContainer - системний контейнер домену
- UsersContainer - контейнер для користувачів домену

Для виведення відомостей про контролер домену, можна скористатися командлетом Get-ADDomainController. За замовчуванням, буде показано інформацію про найближчий контролер домену:

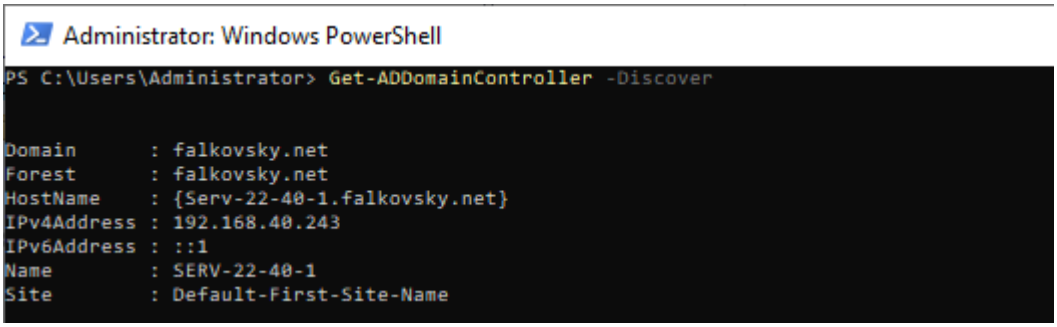


Якщо контролерів домену декілька, переглянути відомості про всі зручно за допомогою параметра -Filter та підстановочних символів. Наприклад, всі контролери домену:

Get-ADDomainController -Filter *

Якщо вжити параметр -Discover, командлет виведе інформацію про найближчий доступний контролер домену:

Get-ADDomainController -Discover Наприклад:



Найближчий доступний у межах того самого сайту контролер у даному випадку визначає процес *DC locator*.

Якщо ви вводите командлет Get-ADDomainController на одному з контролерів домену, то можна дати вказівку обрати як найближчий інший контролер домену, а не той, з якого було запущено команду. Для цього треба додати параметр -AvoidSelf:

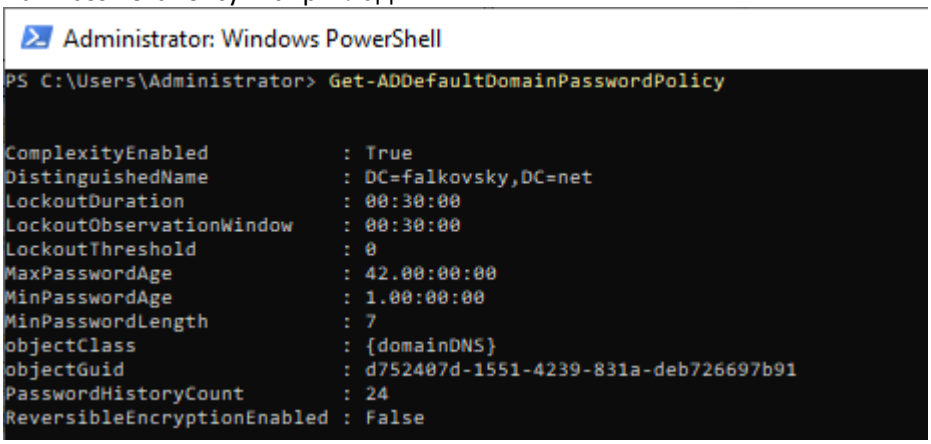
Get-ADDomainController -Discover -AvoidSelf

Вивести інформацію про конкретний контролер домену можна також вказавши його ім’я:

PS C:\> Get-ADDomainController -Identity *DCName*

де *DCName* - ім’я контролера домену.

Групові політики, які стосуються паролів, у багатьох випадках встановлюються на рівні домену. Тому для їх перегляду можна скористатися окремим командлетом Get-ADDefaultDomainPasswordPolicy. Наприклад:



Завдання №10.1

1. Запустіть командну оболонку PowerShell на одному з контролерів вашого домену. Засобами оболонки PowerShell організуйте виведення на екран наступних відомостей:

- ❖ Інформація про поточний домен
- ❖ Відомості про поточний контролер домену
- ❖ Відомості про найближчий контролер домену (але не той, з якого ви вводите команду)
- ❖ Відомості про всі контролери, доступні в межах даного домену (одним командлетом)
- ❖ Інформація про будь-який із наявних у вас контролерів домену за його ім’ям
- ❖ Наявні налаштування групових політик щодо паролів у домені

2. Спираючись на інформацію, одержану під час виконання п. 1, **дайте відповіді на наступні запитання. Поряд вкажіть команди, на основі яких ви одержали ці відповіді.**

- ❖ Яке ім’я поточного домену?
- ❖ Який SID присвоєно цьому домену?
- ❖ Чи є у цього домену дочірні домени?
- ❖ Яке ім’я поточного лісу?
- ❖ Які імена в контролерів домену? Які IP-адреси цих контролерів?
- ❖ Яка операційна система працює на контролерах домену (редакція, версія)? **Не плутати з полем *DomainMode*! (там може фігурувати ім’я іншої версії)**
- ❖ Які контролери домену беруть участь у реплікації?
- ❖ У межах якого сайту працюють контролери домену?
- ❖ Чи є у домені контролери ReadOnly (лише для читання)?
- ❖ Яка мінімальна довжина паролю встановлена груповими політиками для даного домену?
- ❖ Яка максимальна кількість спроб дається на введення паролю?

Звіт має містити скриншоти одержаних відомостей, опис основних дій, відповіді на поставлені запитання.

Робота зі структурою домену

Далі в методичних матеріалах усі приклади та скріншоти наведено на основі Windows Server 2019. Однак методика повністю застосовна і до Server 2022, оскільки обидві версії використовують однакову версію PowerShell за замовчуванням — Windows PowerShell 5.1. Усі команди, сценарії та модулі, що використовуються в цій методиці, підтримуються без змін у Server 2022.

Загальну інформацію про структуру домену (ім'я, батьківський та дочірні домени, SID домену, окремі контейнери та організаційні одиниці тощо) можна почерпнути з виводу командлета Get-ADDomain, описаного вище у цій лабораторній роботі.

Однак для більш ґрунтовної роботи зі структурою домену потрібно мати детальніше уявлення про неї.

Переглянути інформацію всі організаційні одиниці домену можна за допомогою командлета Get-ADOrganizationalUnit. Командлет дає доступ до багатьох відомостей про організаційні одиниці (детальніше - у *Microsoft Docs*), і у багатьох випадках є сенс ці відомості відформатувати та відфільтрувати. Так, у прикладі нижче виведено інформацію про всі організаційні одиниці домену, однак із доступних полів обрано тільки скорочені та повні імена цих одиниць:

```
PS C:\Users\Administrator> Get-ADOrganizationalUnit -Filter 'Name -like "*"' | Format-Table Name, DistinguishedName -A

Name               DistinguishedName
----               -----------------
Domain Controllers OU=Domain Controllers,DC=holovnia,DC=net
Branch-19-30-1     OU=Branch-19-30-1,DC=holovnia,DC=net
Branch-19-30-2     OU=Branch-19-30-2,DC=holovnia,DC=net
Branch-19-30-3     OU=Branch-19-30-3,DC=holovnia,DC=net
Dep-19-30-1-1      OU=Dep-19-30-1-1,OU=Branch-19-30-1,DC=holovnia,DC=net
Dep-19-30-1-2      OU=Dep-19-30-1-2,OU=Branch-19-30-1,DC=holovnia,DC=net
Dep-19-30-1-3      OU=Dep-19-30-1-3,OU=Branch-19-30-1,DC=holovnia,DC=net
B19301-Admins      OU=B19301-Admins,OU=Branch-19-30-1,DC=holovnia,DC=net
Dep-19-30-2-1      OU=Dep-19-30-2-1,OU=Branch-19-30-2,DC=holovnia,DC=net
Dep-19-30-3-1      OU=Dep-19-30-3-1,OU=Branch-19-30-3,DC=holovnia,DC=net
B19302-Admins      OU=B19302-Admins,OU=Branch-19-30-2,DC=holovnia,DC=net
Computers-19-30    OU=Computers-19-30,DC=holovnia,DC=net
Comp-Branch-19-30-1 OU=Comp-Branch-19-30-1,OU=Computers-19-30,DC=holovnia,DC=net
```

За повними іменами видно, що окремі організаційні одиниці містяться всередині інших організаційних одиниць. Наприклад, OU Dep-19-30-1-1 міститься всередині OU Branch-19-30-1:

```
Dep-19-30-1-1       OU=Dep-19-30-1-1,OU=Branch-19-30-1,DC=holovnia,DC=net
```

У наступному прикладі додано маску для імен організаційних одиниць - буде виведено лише ті з них, імена яких містять буквосполучення Br:

```
PS C:\Users\Administrator> Get-ADOrganizationalUnit -Filter 'Name -like "Br*"' | Format-Table Name, DistinguishedName -A

Name           DistinguishedName
----           -----------------
Branch-19-30-1 OU=Branch-19-30-1,DC=holovnia,DC=net
Branch-19-30-2 OU=Branch-19-30-2,DC=holovnia,DC=net
Branch-19-30-3 OU=Branch-19-30-3,DC=holovnia,DC=net
```

З виводу можна бачити, що всі ці організаційні одиниці містяться безпосередньо в домені *holovnia.net*:

```
Name           DistinguishedName
----           -----------------
Branch-19-30-1 OU=Branch-19-30-1,DC=holovnia,DC=net
Branch-19-30-2 OU=Branch-19-30-2,DC=holovnia,DC=net
Branch-19-30-3 OU=Branch-19-30-3,DC=holovnia,DC=net
```

Командлет New-ADOrganizationalUnit дає змогу створювати нові організаційні одиниці.

Для створення нової організаційної одиниці безпосередньо на рівні домену вкажіть ім'я цієї організаційної одиниці:

New-ADOrganizationalUnit -Name *OUName*

Наприклад:

```
PS C:\Users\Administrator> New-ADOrganizationalUnit -Name Branch-19-30-4
```

Переконаймося, що нову організаційну одиницю справді створено:

```
PS C:\Users\Administrator> Get-ADOrganizationalUnit -Filter 'Name -like "Br*"' | Format-Table Name, DistinguishedName -A

Name           DistinguishedName
----           -----------------
Branch-19-30-1 OU=Branch-19-30-1,DC=holovnia,DC=net
Branch-19-30-2 OU=Branch-19-30-2,DC=holovnia,DC=net
Branch-19-30-3 OU=Branch-19-30-3,DC=holovnia,DC=net
Branch-19-30-4 OU=Branch-19-30-4,DC=holovnia,DC=net
```

Якщо треба створити нову організаційну одиницю всередині іншої організаційної одиниці, треба вказати шлях.

Наприклад, щоб всередині OU *Branch-19-30-4* створити OU *Dep-19-30-4-1*, скористаємося командою:

```
PS C:\Users\Administrator> New-ADOrganizationalUnit -Name Dep-19-30-4-1 -Path 'OU=Branch-19-30-4,DC=holovnia,DC=net'
```

З'явилася нова організаційна одиниця *Dep-19-30-4-1*, і вона справді є дочірньою для OU *Branch-19-30-4*:

```
PS C:\Users\Administrator> Get-ADOrganizationalUnit -Filter 'Name -like "Dep*"' | Format-Table Name, DistinguishedName -A

Name           DistinguishedName
----           -----------------
Dep-19-30-1-1 OU=Dep-19-30-1-1,OU=Branch-19-30-1,DC=holovnia,DC=net
Dep-19-30-1-2 OU=Dep-19-30-1-2,OU=Branch-19-30-1,DC=holovnia,DC=net
Dep-19-30-1-3 OU=Dep-19-30-1-3,OU=Branch-19-30-1,DC=holovnia,DC=net
Dep-19-30-2-1 OU=Dep-19-30-2-1,OU=Branch-19-30-2,DC=holovnia,DC=net
Dep-19-30-3-1 OU=Dep-19-30-3-1,OU=Branch-19-30-3,DC=holovnia,DC=net
Dep-19-30-4-1 OU=Dep-19-30-4-1,OU=Branch-19-30-4,DC=holovnia,DC=net
```

Вивести всі організаційні одиниці, які містяться всередині даної, можна за допомогою параметра -SearchBase:

```
PS C:\Users\Administrator> Get-ADOrganizationalUnit -LDAPFilter '(name=*)' -SearchBase 'OU=Branch-19-30-4,DC=holovnia,DC=net' -SearchScope OneLevel | Format-Table Name

Name
----
Dep-19-30-4-1
```

Створити нового користувача домену можна за допомогою командлета New-ADUser. У цього командлета дуже багато параметрів (детальніше - у *Microsoft Docs*). Скористаймося деякими з них:

New-ADUser -Name *Логін* -UserPrincipalName *Логін@доменне.ім_я*

-GivenName *Ім_я* -Initials *Ініціали* -Surname *Прізвище*

-Path *Повний_шлях_до_організаційної_одиниці* Наприклад:

```
PS C:\Users\Administrator> New-ADUser -Name "dep41-poa" -UserPrincipalName "dep41-poa@holovnia.net" -GivenName
 "Oleh" -Initials "A." -Surname "Pinchuk" -Path "OU=Dep-19-30-4-1,OU=Branch-19-30-4,DC=holovnia,DC=net"
```

Вивести відомості про користувача можна командлетом Get-ADUser. Наприклад, так:

```
PS C:\Users\Administrator> Get-ADUser -Identity dep41-poa -Properties *
```

Оскільки для параметра -Properties задано маску *, то буде показано всі властивості облікового запису, і вивід стане доволі довгим. Проте в ньому можна помітити, що наразі обліковий запис відключений:

```
DistinguishedName                 : CN=dep41-poa,OU=Dep-19-30-4-1,OU=Branch-19-30-4,DC=holovnia,DC=net
Division                          :
DoesNotRequirePreAuth             : False
dSCorePropagationData             : {01.01.1601 2:00:00}
EmailAddress                      :
EmployeeID                        :
EmployeeNumber                    :
Enabled                           : False
Fax                               :
GivenName                         : Oleh
HomeDirectory                     :
HomedirRequired                   : False
HomeDrive                         :
HomePage                          :
```

Також для обліковки не задано пароль. На рівні властивостей це виглядає так: пароль ніколи не було встановлено, але термін його дії скінчився:

```
PasswordExpired                         : True
PasswordLastSet                         :
PasswordNeverExpires                    : False
PasswordNotRequired                     : False
POBox                                   :
```

Так відбувається тому, що пароль є обов'язковим (окрім тих випадків, коли властивість *PasswordNotRequired* має значення *True*). Задамо новоствореному користувачу тимчасовий пароль:

```
PS C:\Users\Administrator> Set-ADAccountPassword -Identity dep41-poa -Reset
Please enter the desired password for 'CN=dep41-poa,OU=Dep-19-30-4-1,OU=Branch-19-30-4,DC=holovnia,DC=net'
Password: ********
Repeat Password: ********
```

Параметр -Reset потрібний для того, аби перед новим паролем не вводити старий пароль, котрого поки не було задано.

Увімкнемо обліковий запис та, оскільки пароль тимчасовий, додамо вимогу обов'язкової зміни паролю під час наступного входу цього користувача:

```
PS C:\Users\Administrator> Set-ADUser -Identity dep41-poa -Enabled $true -ChangePasswordAtLOgon $true
```

Тепер обліковий запис увімкнено:

```
PS C:\Users\Administrator> Get-ADUser -Identity dep41-poa -Properties Enabled


DistinguishedName : CN=dep41-poa,OU=Dep-19-30-4-1,OU=Branch-19-30-4,DC=holovnia,DC=net
Enabled           : True
GivenName         : Oleh
Name              : dep41-poa
ObjectClass       : user
ObjectGUID        : 9413fa0a-bd49-4a2f-a463-f21ed849fcff
SamAccountName    : dep41-poa
SID               : S-1-5-21-3806899930-2341899199-3056094349-1123
Surname           : Pinchuk
UserPrincipalName : dep41-poa@holovnia.net
```

За замовчуванням облікові записи додаються до групи *Domain Users*, і нас це влаштовує.

Якщо потрібно додати обліковий запис до іншої групи, можна скористатися командлетами New-AddGroup (створення нової групи) та AddADGroupMember (додавання нових членів групи). Більше про ці командлети є в *Microsoft Docs*.

Командлет Search-ADAccount також надає можливості для пошуку облікових записів користувачів, комп'ютерів та служб за певною ознакою.

Наприклад, якщо треба відшукати облікові записи, котрі було заблоковано, введемо наступну команду:

```
PS C:\Users\Administrator> Search-ADAccount -LockedOut | Format-Table Name,ObjectClass -A

Name            ObjectClass
----            -----------
Larysa P. Kotyk user
```

Командлет Search-ADAccount має ще багато корисних параметрів. Більше про доступні параметри та приклади їх використання на практиці - у *Microsoft Docs*.

Щоб розблокувати користувача, нам знадобиться його логін (поле *SamAccountName*). Вивести потрібну інформацію можна, наприклад, так:

```
PS C:\Users\Administrator> Get-ADUser -Filter 'Name -like "*Kotyk"' | Format-Table Name,SamAccountName -A

Name            SamAccountName
----            --------------
Larysa P. Kotyk dep11-klp
```

Тепер розблокуємо користувача за допомогою командлета UnlockADAccount, вказавши логін користувача як значення параметра *-Identity*:

```
PS C:\Users\Administrator> Unlock-ADAccount -Identity dep11-klp
```

Переконаймося, що такого заблокованого облікового запису більше немає:

```
PS C:\Users\Administrator> Search-ADAccount -LockedOut | Format-Table Name,ObjectClass -A
```

Після цього може знадобитися змінити користувачу пароль та зобов'язати його змінити пароль під час наступного входу.

Видалення облікового запису користувача робиться командлетом Remove-ADUser. Це сама та ситуація, коли незайве застосувати режим прототипування (параметр *-WhatIf*). Так зменшується ризик помилково вилучити не той обліковий запис:

```
PS C:\Users\Administrator> Remove-ADUser -Identity dep41-poa -WhatIf
What if: Performing the operation "Remove" on target "CN=dep41-poa,OU=Dep-19-30-4-1,OU=Branch-19-30-4,DC=holovnia,DC=net".
```

Якщо все правильно, повторюємо команду без параметра *-WhatIf*:

```
PS C:\Users\Administrator> Remove-ADUser -Identity dep41-poa

Confirm
Are you sure you want to perform this action?
Performing the operation "Remove" on target "CN=dep41-poa,OU=Dep-19-30-4-1,OU=Branch-19-30-4,DC=holovnia,DC=net".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "Y"): y
```

Зверніть увагу, що комнадлет Remove-ADUser за замовчуванням працює в режимі підтвердження, хоча параметра *-Confirm* і не вказано.

Переконаймося, що користувача справді видалено за допомогою командлета Get-ADUser. Уважно читайте повідомлення, які у даному випадку виводить вам команда. Якщо команду введено правильно, але такого користувача не знайдено, вгорі виводу буде відповідна інформація:

```
PS C:\Users\Administrator> Get-ADUser -Identity dep41-poa -Properties *
Get-ADUser : Cannot find an object with identity: 'dep41-poa' under: 'DC=holovnia,DC=net'.
At line:1 char:1
+ Get-ADUser -Identity dep41-poa -Properties *
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (dep41-poa:ADUser) [Get-ADUser], ADIdentityNotFoundException
    + FullyQualifiedErrorId : ActiveDirectoryCmdlet:Microsoft.ActiveDirectory.Management.ADIdentityNotFoundExceptio
   n,Microsoft.ActiveDirectory.Management.Commands.GetADUser
```

<u>Завдання №10.2</u>

1. Засобами PowerShell створіть у корені домену організаційну одиницю *BranchGG-NN-4*.
2. Усередині OU *Branch-GG-NN-4* через PowerShell створіть дві дочірні OU: *DepGG-NN-4-1* та *Dep-GG-NN-4-2.*
3. Засобами PowerShell створіть в OU *Dep-GG-NN-4-1* та *Dep-GG-NN-4-2* по одному користувачу. Під час іменування користувачів дотримуйтеся тих самих правил, які ви обрали для імен користувачів у лабораторній роботі №4, коли створювали структуру домену.

<u>Наступні дії також виконайте засобами PowerShell (окрім спроб входу користувачів).</u>

4. Організуйте виведення відомостей про новостворених користувачів - окремо про першого, окремо про другого.
5. Забезпечте двом новостворениим користувачам можливість входу в систему та переконайтеся, що обидва вони можуть увійти з довільної побочої станції домену.
6. Заблокуйте обліковий запис першого користувача шляхом кількаразового введення ним неправильного паролю.

7. Здійсніть пошук облікових записів домену, котрі зараз заблоковано.
8. Розблокуйте обліковий запис першого користувача. Задайте йому новий тимчасовий пароль та у примусовому порядку зобов'яжіть змінити пароль під час наступного входу.
9. Видаліть облікові записи першого та другого користувачів у режимі прототипування.
10. Видаліть обліокві записи першого та другого користувачів насправді. Переконайтеся, що облікові записи справді видалено.

Звіт має містити скриншоти зроблених налаштувань та опис основних дій.

Дослідження налаштувань DNS

Наведемо також окремі команди для дослідження налаштувань DNS.

Щоб вивести всі DNS-записи типу A, введіть командлет GetDnsServerResourseRecord з параметром -RRType "A":

```
PS C:\Users\Administrator> Get-DnsServerResourceRecord -ZoneName "holovnia.net" -RRType "A"

HostName                  RecordType Type       Timestamp           TimeToLive      RecordData
--------                  ---------- ----       ---------           ----------      ----------
@                         A          1          22.11.2021 17:00:00 00:10:00        220.19.30.4
@                         A          1          22.11.2021 17:00:00 00:10:00        220.19.30.3
DomainDnsZones            A          1          22.11.2021 17:00:00 00:10:00        220.19.30.3
ForestDnsZones            A          1          22.11.2021 17:00:00 00:10:00        220.19.30.3
router                    A          1          0                   01:00:00        220.19.30.1
WIN-8T2CASENUDU           A          1          22.11.2021 17:00:00 00:20:00        220.19.30.4
win-sdj836u8svc           A          1          0                   01:00:00        220.19.30.3
WS-19-30-5                A          1          15.11.2021 16:00:00 00:20:00        220.19.30.5
```

Аналогічно можна переглянути записи інших типів - наприклад, записи про сервери імен (тип *NS*):

```
PS C:\Users\Administrator> Get-DnsServerResourceRecord -ZoneName "holovnia.net" -RRType "NS"

HostName                  RecordType Type       Timestamp           TimeToLive      RecordData
--------                  ---------- ----       ---------           ----------      ----------
@                         NS         2          0                   01:00:00        win-sdj836u8svc.holovnia.net.
_msdcs                    NS         2          0                   01:00:00        win-sdj836u8svc.holovnia.net.
```

Перевірити доступність DNS-сервера можна командлетом TestDnsServer за IP-адресою:

```
PS C:\Users\Administrator> Test-DnsServer -IPAddress 220.19.30.3

IPAddress    Result  RoundTripTime TcpTried UdpTried
---------    ------  ------------- -------- --------
220.19.30.3 Success 00:00:11      True     True
```

Завдання №10.3

1. Засобами PowerShell організуйте виведення відомостей про DNS-записи типів A, AAAA, та NS у зоні прямого перегляду.
2. За допомогою PowerShell перевірте наявність доступних DNS-серверів за IPадресами першого та другого контролерів домену. Поясніть одержані результати.

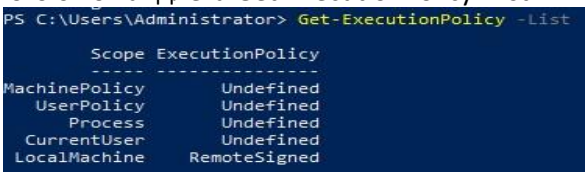
Звіт має містити скриншоти зроблених налаштувань, опис основних дій, а також відповіді на поставлені запитання.

СТВОРЕННЯ ПРОСТИХ СКРИПТІВ POWERSHELL

Налаштування політики виконання скриптів

На робочих станціях та в локальних Windows-системах, які не входять до складу домену, запуск скриптів у системі за замовчуванням не дозволено. Водночас, у Windows Server, навпаки, стандартною є політика виконання *RemoteSigned* для області *LocalMachine*.

Перевірити, якими є поточні налаштування політик виконання для скриптів у вашій системі, можна за допомогою командлета Get-ExecutionPolicy -List:



Детальніше про створення та запуск скриптів - у матеріалах лекції на відповідну тему.

Завдання №10.4

1. На контролері домену створіть та протестуйте скрипт, який виводить на екран повідомлення *"Hello world!"*.

2. Використовуючи команди, які ви виконували у межах завдань Напишіть скрипт lab8-*GG-NN*.ps1, який виконує наступні дії:

- ❖ виводить відомості про домен загалом, про найближчий контролер домену, а також про найближчий контролер домену, але не той, з якого запущено скрипт;
- ❖ виводить інформацію про наявність доступних DNS-серверів за IP-адресами першого та другого контролерів домену;
- ❖ створює у корені домену організаційну одиницю *Branch-GG-NN-5;*
  - всередині OU *Branch-GG-NN-5* створює дочірню OU *Dep-GG-NN-5-1;*
- ❖ в OU *Dep-GG-NN-5-1* створює обліковий запис користувача, задає йому тимчасовий пароль, забезпечує можливість входу і зобов'язує користувача змінити тимчасовий пароль під час першого входу.

Під час іменування користувачів дотримуйтеся тих самих правил, які ви обрали для імен користувачів, коли створювали структуру домену.

3. Перевірте роботу скрипта, запустивши його на контролері домену і переглянувши результати його роботи.

Звіт має містити скриншоти використаних команд, скриншот скрипта та одержаних виводів, **а також програмний код скрипта** lab10- *GG-NN* **(окремим файлом, прикріпленим до листа - розширення приберіть для уникнення проблем з вкладанням)**.

Вимоги до звіту

Звіт має містити інформацію відповідно до завдань 10.1-10.4.

Рекомендовані джерела

1) ActiveDirectory Module - Microsoft Docs. URL: https://docs.microsoft.com/en-us/powershell/module/activedirectory/?view=windowsserver2019-ps
2) E. Wilson. Windows PowerShell 3.0 Step by Step. - Microsoft, 2013.
3) M. Henderson, J. Krause. Windows Server 2019 Cookbook Second Edition (Second Edition). Packt Publishing, 2020 (p. 82).