1. Министерство образования и науки Российской Федерации
2. Санкт-Петербургский Политехнический Университет Петра Великого
3. —
4. Институт компьютерных наук и технологий
5. **Кафедра «Информационная безопасность компьютерных систем»**

**ЛАБОРАТОРНАЯ РАБОТА № 2**

1. «**Установка и настройка службы каталога Active Directory**»
2. по дисциплине «Безопасность современных информационных технологий»
3. Выполнил
4. студент гр. 33508/3 Мальцев К. И.

<*подпись*>

2. Преподаватель Иванов Д. В.

<*подпись*>

1. Санкт-Петербург
2. 2018

# Формулировка задания

Цель работы — изучить способы и механизмы установки и настройки службы каталога Active Directory.

Задачи:

1. Придумать и описать структуру некоторой компании (без привязки к компьютерному обеспечению).
2. Разработать логическую структуру Active Directory, удовлетворяющую структуре компании.
3. Выполнить установку и настройку Active Directory в соответствии с разработанной структурой.
4. Создать следующие административные группы пользователей в Active Directory: Account Managers, Help Desk, Resource Admins, Genereal Admins.
5. Настроить полномочия для созданных групп пользователей:

* Account Managers могут только создавать учетные записи для новых сотрудников и удалять их при необходимости.
* Help desk могут только разблокировать пользовательские учетные записи в случае их блокировки при неудачном вводе пароля.
* Resource Admins должны управлять доступом к информационным ресурсам организации.
* General Admins должны обладать полным доступом и контролировать работу администраторов нижних уровней.

1. Создать тестовых пользователей в каждой из групп.
2. Выполнить проверку правильности настройки полномочий путем попыток исполнения соответствующих действий от имен пользователей различных групп.
3. Создать файловый сервер. На файловом сервере должны быть папки для каждого компьютера и каждого пользователя в организации. При входе пользователя на некоторый компьютер, ему должны монтироваться два разделяемых ресурса, соответствующие папке компьютера и папке пользователя на файловом сервере.
4. Создать инструментарий, помогающий в работе администраторам: скрипт для поиска пользователей, у которых скоро истекает срок действия пароля, и скрипт для разблокирования всех учетных записей пользователей в заданном организационном подразделении.

Требования:

* У каждого студента должна быть своя структура организации, повторов быть не может.
* Создание групп пользователей, настройка их полномочий, создание тестовых пользователей и другие настройки должны выполняться автоматизированно с помощью скриптов.
* Логическая структуру Active Directory должна включать в себя, как мимнимум, домены и организационные подразделения (минимум два).
* Демонстрационный стенд должен состоять минимум из трех компьютеров (возможно использование виртуальных машин).
* В состав стенда должны входить сервера на базе Windows 2008 или выше и рабочие станции на базе Windows 7 или выше.
* Количество дней до истечения срока действия пароля для скрипта поиска пользователей, должно являться входным параметром скрипта.
* Название организационного подразделения для скрипта разблокирования учетных записей пользователей должно являться входным параметром скрипта.

# Результаты работы

В ходе работы была придумана и описана структура университета. Университет состоит из трех подразделений: Faculty\_Humanities (гуманитарный факультет), Faculty\_Engineering (технический факультет) и Administration (Администрация). В свою очередь, факультеты содержат свои подразделения (кафедры): Faculty\_Humanities - Department\_Jurisprudence, Department\_Economics; Faculty\_Engineering - Department\_Physics, Department\_Math. На каждой кафедре есть два подразделения: Teachers (Преподаватели), Students (Студенты).

В качестве серверной ОС выступала Windows Server 2012 R2 x64. Первым дело была установлена служба каталогов Active Directory. Далее был создан лес university.com.

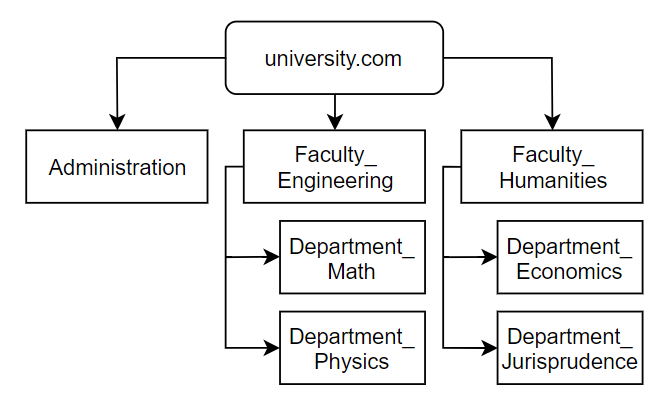


Рис 1. Графическое представление структуры университета.

С помощью скриптов были созданы разделы, а также созданы следующие административные группы: Account Managers, Help Desk, Resource Admins, Genereal Admins. На рисунке 2 изображены административные группы в разделе Administration.

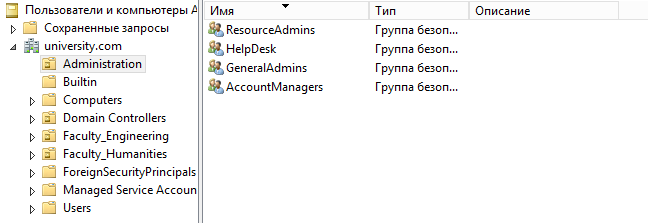


Рис. 2. Созданные OU и административные группы.

С помощью утилиты командой строки DSACLS.EXE и команды dsmod настроены полномочия для созданных групп пользователей:

dsacls.exe ou=Administration,dc=university,dc=com /I:T /G "university\AccountManagers":CC;user

dsacls.exe ou=Administration,dc=university,dc=com /I:S /G "university\AccountManagers":GRGW;;user

dsacls.exe ou=Administration,dc=university,dc=com /I:S /G "university\AccountManagers":CA;"Reset Password";user

dsacls.exe ou=Administration,dc=university,dc=com /I:S /G "university\AccountManagers":RPWP;pwdLastSet;user

dsacls.exe ou=Administration,dc=university,dc=com /I:S /G "university\HelpDesk":RPWP;lockoutTime;user

dsmod group "cn=IIS\_IUSRS,cn=Builtin,dc=university,dc=com" -addmbr "cn=ResourceAdmins,ou=Administration,dc=university,dc=com"

dsmod group "cn=Администраторы,cn=Builtin,dc=university,dc=com" -addmbr "cn=GeneralAdmins,ou=Administration,dc=university,dc=com"

Для каждой группы были созданы тестовые пользователи: Resource\_user, Help\_user, General\_user, Account\_user. Пример групп и их пользователей представлен на рисунке 3.

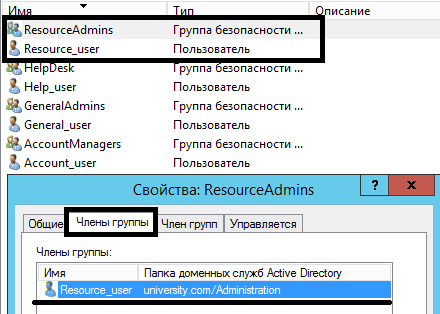


Рис. 3. Тестовые пользователи для каждой группы.

Создадим пользователя “kostya” с помощью разработанного скрипта. В качестве параметров скрипт принимает имя пользователя, пароль, подразделение и команду (создание или удаление). На рисунке 4 изображен пример создания нового пользователя в разделе “Administration”, а на рисунке 5 – его удаление.

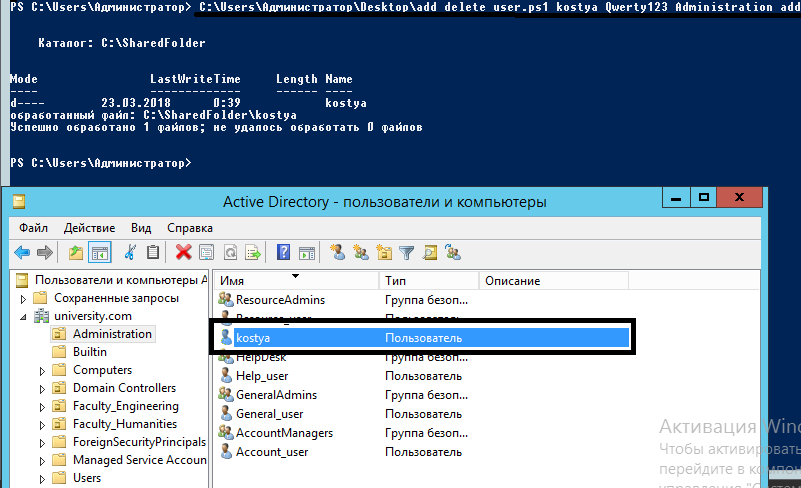


Рис. 4. Пример создания пользователя с помощью скрипта.

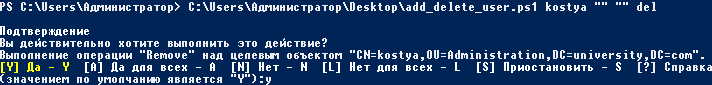


Рис. 5. Пример удаления пользователя с помощью скрипта.

Теперь проверим работоспособность скрипта по разблокировке пользователей. Создадим пользователя “kostya” в разделе Faculty\_Engineering/Department\_math/Students. Затем, настроим политику безопасности Windows таким образом, чтобы при трех неудачных попытках ввода пароля при входе система бы блокировала этого пользователя. Чтобы это сделать, необходимо зайти в Редактор управления групповыми политиками – Default Domain Policy – Параметры – Конфигурация компьютера – Политики – Конфигурация Windows – Параметры безопасности – Политики учётных записей – Политика блокировки учётных записей. Изменение порогового значения показано на рисунке 6.

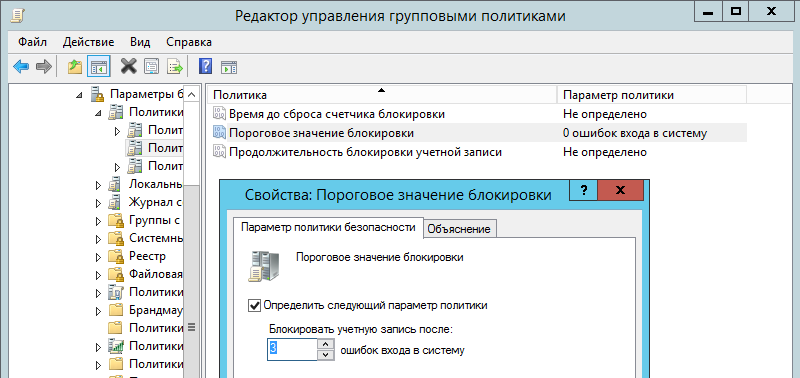


Рис. 6. Изменение порогового значения блокировки.

Далее, попытаемся ввести три раза неверный пароль, после чего, на четвертый раз, система выдает следующее сообщение (рисунок 7):

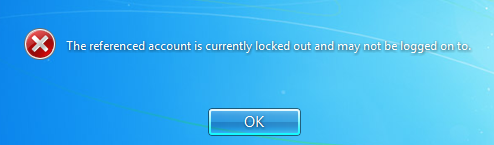


Рис. 7. Сообщение о блокировке.

С помощью скрипта разблокируем пользователя “kostya” (рисунок 8):



Рис. 8. Запуск скрипта для разблокировки пользователя.

С помощью другого скрипта определим, у кого из пользователей заканчивается срок дейсвтия пароля. Результат работы скрипта представлен на рисунке 9.

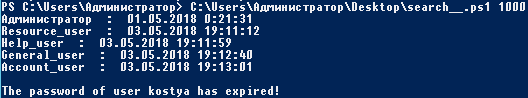


Рис. 9. Пользователи с истекающим сроком действия пароля.

Также в системе был создан общий сетевой диск :Y. Зайдем с пользователя “kostya” и проверим, виден ли диск другим пользователям. На рисунке 10 можно увидеть, что диск виден.

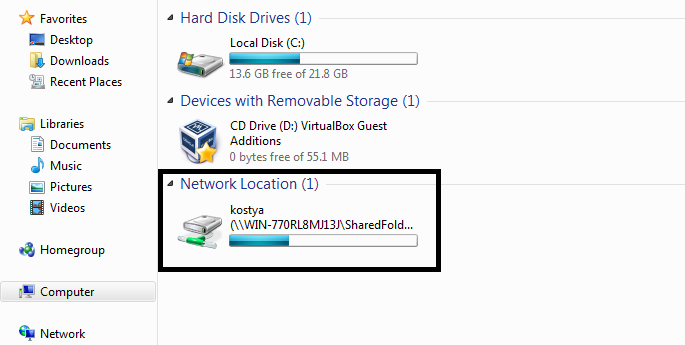


Рис 10. Общий сетевой диск.

# ВЫвОД

В результате работы были изучены способы и механизмы установки и настройки службы каталога Active Directory. Придумана и описана структура университета, разработана логическая структура Active Directory для данной организации. Созданы необходимые административные группы пользователей, файловый сервер (с папками для каждого компьютера и пользователя в организации), тестовые пользователи, показывающие правильность работы настроенной службы каталога AD. Также создан инструментарий, способный искать пользователей с истекающим сроком действия пароля и разблокирования учетных записей в заданном разделе.

# Приложение

**Создание и удаление пользователя:**

param(

[string]$name = $(throw "Error: Enter name."),

[string]$password = $(throw "Error: Enter password."),

[string]$unit = $(throw "Error: Enter unit."),

[string]$command = $(throw "Error: Enter command (add / del)."),

[string]$groupname = ""

)

if($command -eq "add")

{

if ($unit)

{

$secpass = $password | ConvertTo-SecureString -AsPlainText -Force

# делим строку на подстроки, разделенные '/'

$array\_of\_units = $unit -split '/'

# строим путь в подраздел

$array\_of\_units | ForEach-Object {$path = "OU=$\_," + $path}

# в конце добавляем домен

$path += "DC=university, DC=com"

# создаем пользователя

New-ADUser `

-Name $name `

-Path $path `

-SamAccountName $name `

-DisplayName $name `

-AccountPassword $secpass `

-ChangePasswordAtLogon $true `

-Enabled $true `

-HomeDrive "Y:" `

-HomeDirectory "\\WIN-770RL8MJ13J\SharedFolder\$name"

mkdir "C:\SharedFolder\$name"

icacls "C:\SharedFolder\$name" /grant "$($name):(OI)(CI)M"

if ($groupname)

{

Add-ADGroupMember -Identity $groupname -Members $name #$user

}

}

}

elseif($command -eq "del")

{

# удаляем пользователя

Remove-ADUser -Identity $name

# удаляем папку пользователя

Remove-Item "C:\SharedFolder\$name"

}

else

{

Write-Host "Invalid command"

}

**Создание групп:**

Import-Module ActiveDirectory

<#

New-ADGroup - Создает группу AD

-Name - Имя группы

-groupScope - Задает область действия

Global - Эти группы видны через лес, но могут содержать

только учетные записи и глобальные группы из

одного домена

-Path - Где создавать группу

-ManagedBy - Указывает пользователя или группу, управляющую объектом

#>

New-ADGroup -Name "GeneralAdmins" -groupScope Global -Path "ou=Administration, dc=university, dc=com"

New-ADGroup -Name "AccountManagers" -groupScope Global -ManagedBy "GeneralAdmins" -Path "ou=Administration, dc=university, dc=com"

New-ADGroup -Name "HelpDesk" -groupScope Global -ManagedBy "GeneralAdmins" -Path "ou=Administration, dc=university, dc=com"

New-ADGroup -Name "ResourceAdmins" -groupScope Global -ManagedBy "GeneralAdmins" -Path "ou=Administration, dc=university, dc=com"

**Создание разделов:**

Import-Module ActiveDirectory

# create main structures of university

New-ADOrganizationalUnit -Name Faculty\_Humanities -Path "dc=university,dc=com"

New-ADOrganizationalUnit -Name Faculty\_Engineering -Path "dc=university,dc=com"

New-ADOrganizationalUnit -Name Administration -Path "dc=university,dc=com"

# create departments in faculties

New-ADOrganizationalUnit -Name Department\_Jurisprudence -Path "ou=Faculty\_Humanities,dc=university,dc=com"

New-ADOrganizationalUnit -Name Department\_Economics -Path "ou=Faculty\_Humanities,dc=university,dc=com"

New-ADOrganizationalUnit -Name Department\_Physics -Path "ou=Faculty\_Engineering,dc=university,dc=com"

New-ADOrganizationalUnit -Name Department\_Math -Path "ou=Faculty\_Engineering,dc=university,dc=com"

# create members

New-ADOrganizationalUnit -Name Teachers -Path "ou=Department\_Jurisprudence,ou=Faculty\_Humanities,dc=university,dc=com"

New-ADOrganizationalUnit -Name Students -Path "ou=Department\_Jurisprudence,ou=Faculty\_Humanities,dc=university,dc=com"

New-ADOrganizationalUnit -Name Teachers -Path "ou=Department\_Economics,ou=Faculty\_Humanities,dc=university,dc=com"

New-ADOrganizationalUnit -Name Students -Path "ou=Department\_Economics,ou=Faculty\_Humanities,dc=university,dc=com"

New-ADOrganizationalUnit -Name Teachers -Path "ou=Department\_Physics,ou=Faculty\_Engineering,dc=university,dc=com"

New-ADOrganizationalUnit -Name Students -Path "ou=Department\_Physics,ou=Faculty\_Engineering,dc=university,dc=com"

New-ADOrganizationalUnit -Name Teachers -Path "ou=Department\_Math,ou=Faculty\_Engineering,dc=university,dc=com"

New-ADOrganizationalUnit -Name Students -Path "ou=Department\_Math,ou=Faculty\_Engineering,dc=university,dc=com"

**Поиск пользователей с истекающим сроком действия пароля:**

# входной параметр

param([int]$number = 1)

# получаем список пользователей всего домена,

# которые включены и имеют срок действия пароля

# PasswordNeverExpires - пароль без срока действия

$users\_expire\_date = &Get-ADUser -filter {Enabled -eq $True -and PasswordNeverExpires -eq $False} `

# –Properties - какие атрибуты пользователя нужно выводить

# SamAccountName - Имя входа пользователя

# msDS-UserPasswordExpiryTimeComputed - Содержит время истечения срока действия текущего пароля пользователя

# PasswordLastSet - Когда последний раз пользователь менял пароль

# Знак '|' в конце - передаем список в Select-Object (в следующий команду)

–Properties "SamAccountName", "msDS-UserPasswordExpiryTimeComputed", "PasswordLastSet" | `

# Выбираем объект из списка с указанными свойствами

# @{...} - ассоциативный массив

# Массив состоит из имени и времени истечения срока действия пароля

Select-Object –Property “SamAccountName",@{Name="ExpiryDate";`

# Получаем время истечения пароля в виде строки и конвертируем

# его в datetime FromFileTime - форматирование в виде MM/dd/yyyy

Expression={[datetime]::FromFileTime($\_."msDS-UserPasswordExpiryTimeComputed")}}

# проходимся по каждому пользователю из списка

foreach($item in $users\_expire\_date)

{

# получаем дату истечения действия пароля в US

$expire\_date = $item.ExpiryDate

# получаем текущую дату в формате US

$current\_date = &Get-Date

# получаем имя пользователя для входа

$DisplayName = $item.SamAccountName

# -gt (Great than) - больше чем

if($current\_date -gt $item.ExpiryDate)

{

Write-Host "The password of user" $DisplayName "has expired!"

Exit

}

# получаем время до истечения действия пароля

$lasts = $expire\_date - $current\_date

# -lt (Less than) - меньше чем

if($lasts.Days -lt $number)

{

Write-Host "The password of user" $DisplayName "expires less then" $Number "days!"

Write-Host "The password of user" $DisplayName "expires: " $expire\_date

Write-Host “ “

}

}

**Разблокирование пользователей:**

# входной параметр

param([string]$unit = $(throw "Error: Enter unit"))

# делим строку на подстроки, разделенные '/'

$array\_of\_units = $unit -split '/'

# строим путь в подраздел

$array\_of\_units | ForEach-Object {$path = "OU=$\_," + $path}

# в конце добавляем домен

$path += "DC=university, DC=com"

# Search-ADAccount - возвращает список учетных записей

# -SearchBase - поиск в заданном подразделении

# -UsersOnly - только пользователи

# -LockedOut - заблокированные

$list = Search-ADAccount -SearchBase $path -UsersOnly -LockedOut

$count = 0

# проходимся по каждой уч. записи из списка

foreach($user in $list)

{

# разблокировка учетной записи по имени

Unlock-ADAccount -Identity $($user.Name)

Write-Host $user.Name " unlocked"

$count++

}

# -eq (Equal) - если равно

if($count -eq 0)

{

Write-Host "There is no locked users"

}