Оглавление

[Введение 2](#_Toc517874795)

[1.Теоретическая глава 4](#_Toc517874796)

[1.1 Способы работы с Active Directory 4](#_Toc517874797)

[1.2 ADSI 4](#_Toc517874798)

[1.3 Провайдер AD 6](#_Toc517874799)

[1.4 Командлеты AD 8](#_Toc517874800)

[1.5 Управление Active Directory 9](#_Toc517874801)

[1.6 Работа с группами 13](#_Toc517874802)

[2. Аналитическая глава 17](#_Toc517874803)

[2.1 Проектирование доменной структуры 17](#_Toc517874804)

[2.2 Определение количества доменов 17](#_Toc517874805)

[2.3 Проектирование корневого домена леса 20](#_Toc517874806)

[2.4 Проектирование иерархии доменов 22](#_Toc517874807)

[2.5 Назначение владельцев домена 24](#_Toc517874808)

[Управление объектами Active Directory 31](#_Toc517874809)

[3. Проектная глава 39](#_Toc517874810)

[4. Технико-экономическое обоснование и расчет экономической эффективности предлагаемого проекта 42](#_Toc517874811)

[4.1 Расчет затрат на разработку и реализацию проекта 43](#_Toc517874812)

[4.2 Расчет текущих расходов в течение срока жизненного цикла проекта 44](#_Toc517874813)

[4.3 Расчет платежей в виде налогов и экономии налогов на прибыль за счет амортизации 46](#_Toc517874814)

[Заключение 48](#_Toc517874815)

[Список литературы 49](#_Toc517874816)

# Введение

Active Directory® - это технология, «внутри» которой многие администраторы проводят большую часть своего времени, выполняя повседневную административную работу – добавление новых пользователей, обновление объектов директории и .т.д. С внедрением в Windows Server® 2008 R2 командлетов, ориентированных на Active Directory, у администраторов появилась возможность экономить массу времени и сил, используя Windows PowerShell® для автоматизации выполнения многих повторяющихся задач, которые раньше отнимали львиную долю времени. Автоматизация также позволяет повысить безопасность и стабильность работы, так как влияние человеческого фактора, а значит, и количество ошибок сводится к минимуму.

В Windows Server 2008 R2 впервые появился модуль Active Directory для Windows PowerShell. Этот модуль установлен на всех контроллерах домена и является частью Remote Server Administration Toolkit (RSAT) для Windows® 7. Модуль содержит командлеты, которые позволяют осуществлять управление Active Directory из командной строки. Также эти командлеты обеспечивают поддержку технологии, которая отвечает за работу новой графической консоли Active Directory Administrative Center. Эти новые командлеты взаимодействуют с веб-сервисом, который является частью Active Directory в Windows Server 2008 R2. Этот же веб-сервис можно добавить в контроллеры доменов Windows Server 2008 и Windows Server 2003, загрузив его с официального сайта Microsoft. Найти его можно здесь:

http://go.microsoft.com/fwlink/?LinkId=193581

Нет необходимости устанавливать этот веб-сервис на все контроллеры домена. Вполне достаточно того, чтобы он был на контроллере домена того сайта, на котором находится административный центр. Точно так же, нет нужды устанавливать сами командлеты на ваши контроллеры домена. Командлеты нужны только на том клиентском компьютере, на котором вы хотите их использовать.

# 1.Теоретическая глава

## 1.1 Способы работы с Active Directory

Директория Active Directory является основой корпоративных сетей на базе Windows Server 2000, 2003 и 2008. Именно там хранятся все учетные записи пользователей, информация о группах, компьютерах сети, ящиках электронной почты и многом другом.

Всем этим богатством надо управлять, для чего предназначен соответствующий инструментарий, входящий в состав Windows Server, но именно PowerShell позволяет легко автоматизировать массовые действия, направленные на большое количество объектов.

Существует три основных способа работы с Active Directory в Windows PowerShell:

* с помощью интерфейса Active Directory Service Interfaces (ADSI) — этот способ является наиболее сложным, но работает в любой установке PowerShell и не требует дополнительных модулей. Он также наиболее близок к способу управления, который использовался в языке сценариев VBScript;
* с помощью провайдера Active Directory, входящего в расширения PowerShell, — этот способ позволяет подключить директорию в виде диска на вашем компьютере и перемещаться по ней с помощью соответствующих команд: dir, cd и т.д. Данный способ требует установки дополнительного модуля с сайта codeplex;
* с помощью командлетов управления Active Directory — это наиболее удобный способ манипулирования объектами директории, но он тоже требует дополнительной инсталляции соответствующих модулей.

## 1.2 ADSI

Active Directory Service Interfaces (ADSI) хорошо знаком всем, кто пытался писать сценарии на языке VBScript. В PowerShell этот интерфейс реализован с помощью так называемого адаптера. Указав в квадратных скобках название адаптера (ADSI) и путь к объекту в директории на языке LDAP-запроса (Lightweight Directory Access Protocol — протокол работы с директориями, который поддерживает и AD), мы получаем доступ к объекту из директории и можем дальше вызывать его методы.

Например, подсоединимся к одному из контейнеров директории и создадим в нем новую пользовательскую учетную запись.

$objOU=[ADSI]”LDAP://mydc:389/ou=CTO,dc=Employees,dc=testdomain,dc=local”

Итак, теперь у нас переменная $objOU содержит информацию о контейнере (имена переменных в PowerShell начинаются со значка доллара).

Вызовем метод *Create* и создадим в контейнере нового пользователя:

$objUser = $objOU.Create(“user”, “cn=Dmitry Sotnikov”)

Теперь мы можем устанавливать различные атрибуты:

$objUser.Put(«sAMAccountName”, «dsotnikov”)

И наконец, укажем директории, что эти изменения надо применить:

$objUser.SetInfo()

Преимуществами использования адаптера ADSI являются:

* его наличие в любой поставке PowerShell. Если у вас установлен PowerShell и есть директория, с которой вам надо работать, — вы имеете все, что вам надо;
* применение подхода, близкого к VBScript. Если у вас богатый опыт работы с директорией на языке сценариев VBScript или в приложениях .NET, вы сможете уверенно себя чувствовать, используя этот подход.

К сожалению, у метода есть и недостатки:

* сложность — это самый сложный способ работы с директорией. Писать путь к объекту в виде запроса LDAP нетривиально. Для любой работы с атрибутами требуется указание их внутренних имен, а значит, надо помнить, что атрибут, обозначающий город пользователя, называется не «City», а «l» и т.д.;
* громоздкость — как видно из примера, простейшая операция создания одной учетной записи занимает как минимум четыре строчки, включая служебные операции подсоединения к контейнеру и применения изменений. Таким образом, даже относительно простые операции становятся похожи на сложные сценарии.

## 1.3 Провайдер AD

PowerShell позволяет представлять различные системы в виде дополнительных дисков компьютера с помощью так называемых провайдеров. Например, в состав поставки PowerShell входит провайдер реестра и мы можем перемещаться по реестру с помощью знакомых и любимых всеми нами команд cd и dir (для любителей UNIX команда ls тоже поддерживается).

Провайдера Active Directory в составе PowerShell нет, но его можно установить, зайдя на сайт проекта расширений PowerShell — PowerShell Community Extensions: <http://www.codeplex.com/PowerShellCX>.

Это проект с открытым кодом, который добавляет большое количество команд в систему PowerShell, а кроме того, устанавливает провайдера AD.

**

Рис. 1.1 Использование провайдера Active Directory

После установки расширений, набрав Get-PSDrive, мы видим, что к прежним дискам добавился диск текущей активной директории.

Теперь мы можем зайти в эту директорию, набрав cd и указав имя домена, а в любом контейнере использовать команду dir, чтобы увидеть его содержимое.

С объектами дальше можно работать так же, как это делалось с применением адаптера ADSI.

Кроме того, можно вызывать и другие привычные команды управления файлами (например, del).

К несомненным преимуществам использования провайдера можно отнести:

естественность представления структуры директории — директория AD по своей природе иерархична и похожа на файловую систему;

удобство нахождения объектов — применять cd и dir куда удобнее, чем составлять запрос на языке LDAP.

Из недостатков бросаются в глаза:

* сложность внесения изменений в объекты — провайдер помогает легко добраться до объекта, но чтобы что-либо поменять, нам опять приходится использовать все те же директорные объекты, что и в методе ADSI, а для этого надо оперировать на низком уровне служебных методов и атрибутов AD;
* необходимость дополнительной установки — провайдер не входит в состав PowerShell, и для его применения необходимо скачать и установить расширения PowerShell;
* третьестороннее происхождение — расширения PowerShell не являются продуктом компании Microsoft. Они созданы энтузиастами проекта. Вы вольны их использовать, но за технической поддержкой придется обращаться не в Microsoft, а на сайт проекта.

## 1.4 Командлеты AD

Кроме описанного выше провайдера, для работы с AD существует и набор командлетов (часто называемых также AD cmdlets или QAD cmdlets), доступный с сайта <http://www.quest.com/activeroles_server/arms.aspx>.

Командлеты состоят из стандартных глаголов операций (get-, set-, rename-, remove-, new-, move-, connect-) и существительных объектов с префиксом QAD (-QADUser, -QADGroup, -QADComputer, -QADObject).

Например, чтобы создать новую четную запись пользователя, понадобится выполнить такую команду:

New-QADUser -ParentContainer scorpio.local/Employees -Name ‘Dmitry Sotnikov’

Преимущества данного подхода таковы:

* простота — использование командлетов скрывает от вас сложность директории, ее схемы и внутренних атрибутов. Вы работаете с объектами директории на уровне понятных названий объектов (user, group, computer), их свойств (name, password, city, department) и действий над ними (get, set, remove, move, new);
* краткость и выразительность — как мы видели, большую часть действий с помощью командлетов можно выразить в виде простых и естественных однострочных операций.

Недостатками такого подхода можно считать:

* необходимость дополнительной установки — командлеты, как и провайдер, не входят в состав PowerShell, и для их использования необходимо скачать и установить соответствующую библиотеку;
* третьестороннее происхождение — командлеты для работы с AD не являются продуктом компании Microsoft. Они созданы партнером Microsoft — компанией Quest Software. Вы вольны их применять, но за технической поддержкой придется обращаться не в Microsoft, а на форумы по работе с Active Directory на сайте PowerGUI.org.

На наш взгляд, данные недостатки с лихвой компенсируются простотой и естественностью в использовании, так что практические примеры будут приведены с применением именно этого подхода.

## 1.5 Управление Active Directory

Давайте посмотрим, как PowerShell позволяет выполнять основные операции по работе с директорией AD:

* получение информации;
* изменение свойств;
* работа с группами;
* создание новых объектов;
* изменение структуры директории

Получение информации осуществляется в PowerShell с помощью командлетов с глаголом Get.

Например, чтобы получить список всех пользователей, наберем:

Get-QADUser

Для групп:

Get-QADGroup

Для записей компьютеров:

Get-QADComputer

Если вам нужны не все записи, а какие-то конкретные, вы можете выбрать именно их с помощью параметров команд.

**

Рис. 1.2 Получение списка пользователей

Все группы из контейнера Users:

Get-QADGroup -SearchRoot scorpio.local/users

Все пользователи из отдела продаж московского офиса, чьи имена начинаются на букву A:

Get-QADUser -City Moscow -Department Sales -Name a\*

При этом вы можете сказать PowerShell’y, в каком виде вы хотите видеть получаемую информацию.

Таблица с именами, городами и подразделениями сотрудников:

Get-QADUser | Format-Table Name, City, Department

То же самое с сортировкой по городам:

Get-QADUser | Sort City | Format-Table DisplayName, City, Department

**

Рис. 1.3 Сортировка значений и выбор полей для вывода

Для списочного представления той же информации просто используем команду Format-List:

Get-QADUser | Format-List Name, City, Department

Экспортировать информацию в файл CSV (comma-separated values — значения через запятую):

Get-QADUser | Select Name, City, Department | Out-CSV users.csv

Создать отчет в формате HTML:

Get-QADUser | Select Name, City, Department | ConvertTo-HTML | Out-File users.html

Таким образом, одной строчкой простой команды PowerShell вы можете создавать сложные отчеты в удобном для вас формате.

**

Рис. 1.4 PowerShell позволяет менять атрибуты множества
записей одной командой

После того как мы освоились с получением информации из директории, пришла пора что-нибудь в ней поменять.

Свойствами объектов можно манипулировать с помощью команд Set-\*.

Например, поменяем мне телефон:

Set-QADUser ‘Dmitry Sotnikov’ -Phone ‘111-111-111’

Но, разумеется, куда более интересны массовые изменения. Для этого мы можем применять конвейер PowerShell, то есть получать список нужных нам объектов с помощью команд Get- и отправлять их в команду Set- для внесения изменений.

Например, наш пермский офис переехал в новое помещение. Возьмем всех пользователей Перми и присвоим им новый номер телефона:

Get-QADUser -City Perm | Set-QADUser -PhoneNumber ‘+7-342-1111111’

Для более сложных манипуляций можно использовать командлет ForEach-Object. Например, каждому пользователю присвоим описание, состоящее из его отдела и города:

Get-QADUser | ForEach-Object { Set-QADUser $\_ -Description (S\_.City + « « + $\_.Department) }

Переменная $\_ в данном примере обозначает текущий объект коллекции.

**

Рис1.5 PowerShell предоставляет возможности удобной работы
с группами пользователей

## 1.6 Работа с группами

Работа с группами и членством в них — еще одна массовая операция, которую часто хочется автоматизировать. PowerShell предоставляет такую возможность.

Получение членов группы производится с помощью командлета Get-QADGroupMember:

Get-QADGroubMember Managers

Добавить объект в группу тоже несложно:

Add-QADGroupMember Scorpio\Managers -Member dsotnikov

Аналогично удаление из группы осуществляется с помощью командлеты Remove-QADGroupMember.

Но, разумеется, наиболее полезными являются массовые манипуляции. Добавим всех менеджеров в соответствующую группу:

Get-QADUser -Title Manager | Add-QADGroupMember Scorpio\Managers

Скопируем членство в группе:

Get-QADGroupMember Scorpio\Managers | Add-QADGroupMember Scorpio\Managers\_Copy

Используем фильтр, чтобы скопировать не всех членов группы, а только тех, кто отвечает определенному критерию (например, находится в нужном регионе):

Get-QADGroupMember Scorpio\Managers | where { $\_.City -eq ‘Ekaterinburg’} | Add-QADGroupMember Scorpio\Ekaterinburg\_Managers

Обратите внимание, как мы отфильтровали пользователей с помощью команды where и логического условия (логический оператор -eq — это оператор равенства в PowerShell, *от англ.*equals).

Создание объектов, как мы уже видели, осуществляется командами New:

New-QADUser -ParentContainer scorpio.local/Employees -Name ‘Dmitry Sotnikov’

New-QADGroup -ParentContainer scorpio.local/Employees -Name ‘Managers’ -Type Security -Scope Global

Вы можете установить и любые другие атрибуты в процессе создания записи:

New-QADUser -ParentContainer scorpio.local/Employees -Name ‘Dmitry Sotnikov’ -samAccountName dsotnikov -City ‘Saint-Petersburg’ -Password ‘P@ssword’

Чтобы активировать запись, просто отправьте ее по конвейеру в Enable-QADUser (не забудьте установить пароль — иначе операция не пройдет):

New-QADUser -ParentContainer scorpio.local/Employees -Name ‘Dmitry Sotnikov’ -Password ‘P@ssword’ | Enable-QADUser

Можно прочитать список пользователей и из файла. Например, если у нас есть файл, в котором через запятую перечислены атрибуты новых пользователей, то мы можем смело отправлять их на создание с помощью Import-CSV:

Import-CSV new\_users.csv | ForEach-Object { New-QADUser -ParentContainer scorpio.local/users -Name ($\_.Familia + ‘, ’ + $\_.Imya) -samAccountName ($\_.Imya[0] + $\_.Familia) -Department $\_.Department -Title $\_.Title}

Обратите внимание на то, что мы на лету составляем название учетной записи из фамилии и имени пользователя.

**

Рис.1.6 Пример использования файла импорта
записей

И наконец, конечно же, можно управлять структурой директории.

Например, можно создавать новые контейнеры:

New-QADObject -type OrganizationUnit -ParentContainer scorpio.local -Name NewOU

и перемещать в них объекты по одному:

Move-QADObject MyServer -To scorpio.local/servers

или оптом:

Get-QADUser -Disabled | Move-QADObject -To scorpio.local/Disabled

**

Рис. 1.7 Импортируем файл и создаем новые учетные записи

**

Рис. 1.8 Мы легко можем выбрать учетные записи, удовлетворяющие
определенному критерию, и переместить их в другой контейнер

Мы рассмотрели только малую часть сценариев по управлению активной директорией. Чтобы получить полный перечень командлетов для AD, выполните команду:

Get-Command \*-QAD\*

Чтобы получить с правку по любой команде:

Get-Help Get-QADUser

Чтобы узнать, какие свойства есть у выдаваемого командой объекта:

Get-User | Get-Member

Возможности PowerShell практически безграничны, но при этом найти их достаточно легко.

# 2. Аналитическая глава

## 2.1 Проектирование доменной структуры

Как только вопрос о количестве развертываемых лесов улажен, необходимо определить доменную структуру в пределах каждого из лесов. Ваша первая задача состоит в том, чтобы задокументировать конфигурацию текущих служб каталога и определить, какая часть текущей инфраструктуры может быть модернизирована, а какая должна быть реструктурирована или заменена. Затем определяется потребное количество доменов и их иерархия.

Домены и проект Active Directory

Домены используются для разделения большого леса на более мелкие компоненты для целей репликации или администрирования. Следующие характеристики домена крайне важны при проектировании Active Directory.

* Граница репликации. Границы домена являются границами репликации для раздела домена каталога и для информации домена, хранящейся в папке Sysvol на всех контроллерах домена. В то время как другие разделы каталога (раздел схемы, конфигурации и GC) реплицируются по всему лесу, раздел каталога домена реплицируется только в пределах одного домена.
* Граница доступа к ресурсам. Границы домена являются также границами для доступа к ресурсам. По умолчанию пользователи одного домена не могут обращаться к ресурсам, расположенным в другом домене, если только им не будут явно даны соответствующие разрешения.
* Границы политики безопасности. Некоторые политики безопасности могут быть установлены только на уровне домена. Эти политики, такие как политика паролей, политика блокировки учетных записей и политика билетов Kerberos, применяются ко всем учетным записям домена.

## 2.2 Определение количества доменов

В то время как большинство компаний развертывает единственный лес, некоторые крупные компании развертывают несколько доменов в пределах этого леса. Проще всего управлять единственным доменом, он обеспечивает пользователей наименее сложной средой. Однако имеется ряд причин для развертывания несколько доменов.

Для большинства компаний идеальный проект Active Directory Windows Server 2003 будет включать множество более мелких доменов, чем имелись до этого в Windows NT. Для некоторых компаний несколько доме-

нов Windows NT могут объединиться в единственный домен Active Directory. Многие из ограничений, которые приводили к необходимости развертывания нескольких доменов в Windows NT, были устранены в Windows Server 2003. Следующие факторы делают развертывание единственного домена реальной возможностью для многих компаний, имеющих несколько доменов в Windows NT.

Важным требованием при проектировании Active Directory является баланс между оптимальной структурой сети и тем, что уже развернуто к настоящему моменту. Всякий раз, когда вы готовитесь к созданию проекта Active Directory, необходимо рассмотреть уже имеющуюся инфраструктуру и последствия перехода к Active Directory. Текущая доменная структура может не быть идеальной для ее обновления до Active Directory. Однако модернизировать ее значительно легче и дешевле, чем создать идеальную структуру Active Directory, а затем переместить все объекты домена в новые домены. Возможно, что вам придется работать не с идеальной структурой Active Directory, потому что вы будете модернизировать текущие домены. Может выясниться, что текущая структура настолько далека от идеальной, что дополнительная работа и стоимость по реструктурированию всех доменов будет оправдана. Вероятно также, что текущая структура окажется почти приемлемой, но вы захотите сделать в ней некоторые изменения. В этом случае вы можете модернизировать один или несколько доменов и затем присоединить другие домены к модернизированным.

Готовясь к проектированию своей службы Active Directory, вы сначала создадите идеальный проект Active Directory, а затем еще один, основанный на оптимальном сценарии обновления текущей среды. Очень вероятно, что ваша окончательная модель окажется где-нибудь посередине.

Взаимодействие между идеальным и реальным проектом иллюстрирует другой важный аспект проектирования Active Directory: проектирование почти всегда представляет собой итерационный процесс. Вы можете начать проектирование, имея в голове одну модель, и по мере сбора дополнительной информации, вам, возможно, потребуется ее изменить. Когда вы начнете тестировать реализацию или сценарии перехода, вероятно, проект Active Directory опять изменится.

Однако важно, чтобы некоторые части вашего проекта приняли окончательные формы прежде, чем вы начнете развертывание. Реализацию

сильно затрагивает решение о количестве лесов и доменов, а также проектирование доменного пространства имен. Эти решения трудно изменить после того, как развертывание началось. Другие решения типа финального проекта OU и проекта сайтов легко изменить после развертывания.

Ограничения на размер базы данных в значительной степени были сняты в Active Directory, теперь она может содержать несколько сотен тысяч объектов. Для всех, кроме самых больших, компаний общее количество объектов в Active Directory не будет превышать возможное количество объектов в домене.

Одна из причин создания дополнительных доменов в Windows NT состояла в том, чтобы ограничивать или делегировать административный доступ. В Active Directory структура OU создает иерархию в пределах домена, которая облегчает делегирование администрирования определенным частям каталога и ограничивает административный доступ.

Если ваша компания часто реорганизуется, и пользователи передвигаются между деловыми подразделениями, то перемещать их между OU в домене достаточно легко. Труднее перемещать пользователей между доменами.

Управлять одним доменом легче в том отношении, что надо заботиться только об одном наборе администраторов доменного уровня и одном наборе политик доменного уровня. Кроме того, вы должны управлять только одним набором контроллеров домена.

Самый легкий сценарий для управления групповыми политиками — это среда отдельного домена. Некоторые компоненты групповой политики хранятся в папке Sysvol на каждом контроллере домена. Если вы имеете только один домен, групповая политика автоматически копируется на все контроллеры домена.

Единственный домен является самой легкой средой для планирования аутентификации и доступа к ресурсам. Имея единственный домен, вам не нужно беспокоиться о доверительных отношениях или о назначении доступа к ресурсам для пользователей из других доменов.

Выбор нескольких доменов

В то время как модель единственного домена может быть идеальной для многих компаний, большинство крупных компаний развертывают несколько доменов. Существует много серьезных оснований для такого решения.

* Трафик репликации должен быть ограничен. Раздел каталога домена, который является самым большим и наиболее часто изменяемым разделом каталога, копируется на все контроллеры домена в домене. В некоторых случаях это может вызывать слишком большой трафик репликации между офисами компании (даже если сконфигурировано несколько сайтов).
* Этот выбор делается, если между офисами компании существуют медленные сетевые подключения или если в офисах имеется много пользователей. Единственный способ ограничить в этом случае трафик репликации состоит в том, чтобы создать дополнительные домены.
* Любые офисы компании, связь между которыми обеспечивается только простым протоколом передачи почты (SMTP), должны конфигурироваться как отдельные домены. Информация домена не может реплицироваться через связи сайта, использующие протокол SMTP.
* Единственный способ иметь различную политику паролей, политику блокировки учетных записей и политику билетов Kerberos состоит в развертывании отдельных доменов.
* Если вам необходимо ограничивать доступа к ресурсам и иметь административные разрешения, вы захотите развернуть дополнительные домены. Для некоторых компаний могут существовать юридические причины для создания отдельных административных подразделений.
* В некоторых случаях дополнительные домены создаются потому, что лучший путь перехода для организации состоит в модернизации нескольких уже имеющихся доменов.

Существуют серьезные основания для создания дополнительных доменов. Однако каждый дополнительный домен увеличивает административные и финансовые издержки. Каждый дополнительный домен требует дополнительных аппаратных средств и дополнительных администраторов. Пользователи будут обращаться к ресурсам через доверительные отношения, что означает большую сложность и потенциально большее количество мест возможного отказа. Пользователи, путешествующие между доменами, должны подтверждать свои права доступа на контроллер домена в своем домашнем домене. Из-за этих дополнительных затрат общее количество доменов должно оставаться настолько малым, насколько это возможно.

## 2.3 Проектирование корневого домена леса

Другое важное решение, которое вы должны принять при проектировании службы Active Directory большой компании, — действительно ли вы должны развернуть назначенный корневой домен (называемый также пустым корнем). *Назначенный корневой домен* (dedicated root domain) -это домен, который выполняет функции корневого домена леса. В этом домене нет никаких учетных записей пользователей или ресурсов, за исключением тех, которые нужны для управления лесом. Лес с назначенным корневым доменом показан на рисунке 2.1.

Для большинства компаний, развертывающих несколько доменов, настоятельно рекомендуется иметь назначенный корневой домен. Корневой домен - это критический домен в структуре Active Directory. Он содержит административные группы уровня леса (группы Enterprise

Admins и Schema Admins) и хозяев операций уровня леса (хозяина именования доменов и хозяина схемы). Кроме того, корневой домен должен быть всегда доступен, когда пользователи входят на другие домены, не являющиеся их домашними доменами, или когда пользователи обращаются к ресурсам, расположенным в других доменах. Корневой домен нельзя заменять, если он разрушен, его нельзя восстановить, вы должны заново построить весь лес.



Рис. 2.1. Лес с назначенным корневым доменом

Назначенным корневым доменом управлять легче, чем корневым доменом, содержащим много объектов. Поскольку база данных каталога будет мала, достаточно просто поддерживать и восстанавливать контроллеры корневого домена. Между контроллерами корневого домена практически нет трафика репликации, так что не сложно расположить контроллеры домена в нескольких офисах компании для гарантии избыточности. Это облегчит также перемещение корневого домена в другое место. Использование назначенного корневого домена облегчает ограничение членства в административных группах уровня леса. Назначенный корневой домен никогда не устаревает, особенно если домену дают групповое (generic) имя.

По этим причинам большинство компаний, выбирающие несколько доменов, должны развертывать назначенный корневой домен. Даже компании, планирующие только один домен, должны рассмотреть преимущества, связанные с развертыванием назначенного корневого домена.

Назначенный корневой домен требует конфигурирования, которое не применяется к другим доменам леса. Поскольку корневой домен содержит хозяина операций леса, контроллеры домена для корневого домена должны быть защищены в максимально возможной степени. Домен леса также содержит группы, которые могут изменять лес и схему. Члены административных групп в корневом домене должны иметь более высокий уровень доверия, чем в случае с любым другим доменом. Вы, вероятно, захотите использовать опцию Restricted Group (Ограниченная группа) в политике Domain Security Policy (Политика безопасности домена) для управления членством этих групп. Конфигурация DNS корневого домена должна быть настолько безопасной, насколько это возможно. Поскольку установка в корневом домене какого-либо дополнительного компьютера маловероятна, вы должны включить безопасные динамические обновления для корневого домена зоны DNS на время инсталляции контроллеров домена, а затем динамические обновления для этой зоны следует отключить.

## 2.4 Проектирование иерархии доменов

Как только проект корневого домена выполнен, нужно определить количество дополнительных доменов и то, и как они впишутся в пространство имен DNS леса. Используйте рекомендации, полученные ранее. Если текущая служба каталога - это служба для сети Windows NT, то для определения количества доменов Windows Server 2003 вы должны исследовать уже имеющуюся структуру доменов.

Многие крупные компании развернули домены Windows NT, используя модели с одним или несколькими хозяевами домена, в которой одни домены содержали учетные записи пользователей и глобальных групп, а другие — ресурсы компании. В некоторых случаях компании имели дюжины доменов с учетными записями и сотни доменов с ресурсами. Часто домены учетных записей были организованы вокруг географических регионов или деловых подразделений, и каждый из них обычно имел один или несколько доменов ресурсов в пределах одного географического региона или делового подразделения. На рисунке 2.2 показан пример того, как может выглядеть конфигурация доменов.

Переходя к Active Directory, эти компании могут значительно уменьшить количество доменов. Обычный путь обновления для многих состоит в модернизации доменов учетных записей. Поскольку домены Active Directory могут содержать значительно больше объектов, в некоторых случаях компания могла бы соединить несколько главных доменов учетных записей в один домен Active Directory. Как только произойдет модернизация доменов учетных записей, можно реструктурировать домены ресурсов, чтобы они стали организационными единицами в домене

Active Directory. Иногда домены ресурсов можно удалить. Например, некоторые компании имели домены ресурсов для инфраструктуры Exchange Server. При переходе организации к Exchange Server 2000 серверы могут быть развернуты в домене Active Directory. Когда последний сервер, на котором выполняется Exchange Server.5, удаляется, домен Exchange можно также удалить. На рисунке 2.3 показан возможный переход для компании, имеющей несколько доменов Windows.



Рис. 2.2. Типичная модель с несколькими хозяевами для доменов учетных записей и ресурсов в системе Windows

При планировании дополнительных доменов в лесу границы домена обычно определяются или географическим местом расположения корпорации, или деловыми подразделениями. В большинстве случаев предпочтительны домены, основанные на географии. Доменную конфигурацию трудно изменять после развертывания, а домены, основанные на географии, вряд ли будут требовать модификации. Кроме того, в большинстве случаев сетевая топология соответствует географической конфигурации, так что если вы будете создавать дополнительные домены для управления трафиком репликации, то домены, основанные на географии, вероятно, будут наилучшим вариантом. Доменный проект, основанный на деловых подразделениях, выбирается только в том случае, если эти подразделения достаточно автономны. Если каждое деловое подразделение управляет своей собственной службой каталога, то домены, основанные на деловых подразделениях, имеют смысл.



Рис. 2.3. Обновление доменов до Active Directory Windows Server 2003

## 2.5 Назначение владельцев домена

Для каждого из доменов, включенных в проект Active Directory, вы должны назначить владельца домена. В большинстве случаев владельцы домена являются администраторами деловых подразделений или географического региона, в котором был определен домен.

*Примечание. Если вы развертываете назначенный корневой домен, владельцами этого домена являются владельцы леса. Реальные функции, выполняемые в назначенном корневом домене — это функции леса, так что имеет смысл владельцев леса назначать также владельцами корневого домена.*

Роль владельца домена состоит в управлении индивидуальным доменом. Эти задачи включают следующее.

* Создание политик безопасности уровня домена. Это включает политику паролей, политику блокировки учетных записей и политику билетов Kerberos.
* Проектирование конфигурации Group Policy (Групповая политика) уровня домена. Владелец домена может проектировать групповую политику для всего домена и делегировать право связывать групповую политику с администратором уровня OU.
* Создание в домене OU-структуры высокого уровня. После создания OU-структуры высокого уровня задача создания подчиненных OU может быть передана администраторам уровня OU.
* Делегирование административных прав в пределах домена. Владелец домена должен установить административную политику уровня домена (включая политики схем именования, проекта групп и т.д.), а затем делегировать права администраторам уровня OU.
* Управление административными группами уровня домена. Как уже говорилось, администраторы в каждом домене должны иметь высокую степень доверия, потому что их действия могут вызывать последствия на уровне леса. Роль владельца домена состоит в ограничении членства административной группы уровня домена и в делегировании административных прав низшего уровня всегда, когда это возможно.

Для использования данных командлетов в вашей сессии PowerShell должен быть загружен специальный модуль взаимодействия с AD — **Active Directory Module for Windows PowerShell**. Данный модуль впервые был представлен в Windows Server 208 R2. В Windows Server 2012 и выше этот модуль включен по умолчанию. На клиентских компьютерах его можно установить и включить в качестве одного из компонентов RSAT. Проверить, загружен ли модуль можно так:

Get-Module -Listavailable



Рис.2.4 Проверка загрузки модуля ActiveDirectory

Как вы видите, модуль ActiveDirectory загружен. Если нет – импортируйте его командой:

Import-Module activedirectory

Полный список команд модуля можно получить так:

Get-Command -Module ActiveDirectory

В модуле всего доступно 147 командлетов, из которых с группами могут работать 11.

Get-Command -Module ActiveDirectory -Name "\*Group\*"

Вот их список:



Рис.2.5 Вывод командлетов Group

* Add-ADGroupMember
* Add-ADPrincipalGroupMembership
* Get-ADAccountAuthorizationGroup
* Get-ADGroup
* Get-ADGroupMember
* Get-ADPrincipalGroupMembership
* New-ADGroup
* Remove-ADGroup
* Remove-ADGroupMember
* Remove-ADPrincipalGroupMembership
* Set-ADGroup

New-ADGroup – создаем новую группу AD

Создадим новую группу в указанном контейнере (OU) Active Directory с помощью команды **New-ADGroup**:

New-ADGroup "TestADGroup" -path 'OU=Groups,OU=Moscow,DC=corp,dc=winitpro,DC=ru' -GroupScope Global -PassThru –Verbose

С помощью атрибута **Description** можно задать описание группы, а с помощью **DisplayName** изменить отображаемое имя.



Рис.2.6 Проверка загрузки модуля ActiveDirectory

Параметром **GroupScope** можно задать один из следующих типов групп:

* 0 = DomainLocal
* 1 = Global
* 2 = Universal

Создать группу распространения можно так:

New-ADGroup "TestADGroup-Distr" -path 'OU=Groups,OU=Moscow,DC=corp,dc=winitpro,DC=ru' -GroupCategory Distribution -GroupScope Global -PassThru –Verbose

Add-AdGroupMember – добавить пользователей в группу AD

Добавить пользователей в группу Active Directory можно с помощью командлета Add-**AdGroupMember**. Добавим в новую группу двух пользователей:

Add-AdGroupMember -Identity TestADGroup -Members user1, user2

Если список пользователей, которых нужно добавить в группу довольно большой, можно сохранить список учетных записей в CSV файл, затем импортировать данный файл и добавить каждого пользователя в группу.

Формат CSV файла такой (список пользователей по одному в строке, имя столбца – users)



Рис.2.7 Файл users.csv

Import-CSV .\users.csv -Header users | ForEach-Object {Add-AdGroupMember -Identity ‘TestADGroup’ -members $\_.users}

Чтобы получить всех членов одной группы (groupA) и добавить их в другую группу (groupB), воспользуйтесь такой командой:

Get-ADGroupMember “GroupA” | Get-ADUser | ForEach-Object {Add-ADGroupMember -Identity “Group-B” -Members $\_}

В том случае, если нужно скопировать в новую группу и членов всех вложенных групп (рекурсивно), нужно воспользоваться такой командой:

Get-ADGroupMember -Identity “GroupA” -Recursive | Get-ADUser | ForEach-Object {Add-ADGroupMember -Identity “GroupB” -Members $\_}

Remove-ADGroupMember – удалить пользователей из группы

Для удаления пользователей из группы AD нужно использовать командует Remove-ADGroupMember. Удалим из группы двух пользователей:

Remove-ADGroupMember -Identity TestADGroup -Members user1, user2

Подтвердите удаление пользователей из группы:

Если нужно удалить из группы пользователей по списку из CSV файла, воспользуйтесь такой командой:

Import-CSV .\users.csv -Header users | ForEach-Object {Remove-ADGroupMember -Identity ‘TestADGroup’ -members $\_.users}

Get-ADGroup – получить информацию о группе AD

Получить информацию о группе поможет командлет **Get-ADGroup**:

Get-ADGroup 'TestADGroup'

Даная команда выводит информацию об основных атрибутах группы (DN, тип группы, имя, SID). Чтобы вывести значение всех атрибутов группы домена AD, выполните такую команду:

Get-ADGroup 'TestADGroup' -properties \*



Рис.2.8 Вывод атрибутов группы

Как вы видите, теперь стали отображаться такие атрибуты, как время создания и модификации группы, описание и т.д.

С помощью командлета Get-ADGroup можно найти все интересующие вас группы по определенному шаблону. Например, нужно найти все группы AD, в имени которых содержится фраза ***admins*** :

Get-ADGroup -LDAPFilter “(name=\*admins\*)” | Format-Table

Get-ADGroupMember – вывести список пользователей группы AD

Вывести на экран список пользователей группы:

Get-ADGroupMember 'TestADGroup'

Чтобы оставить в результатах только имена пользователей, выполните:

Get-ADGroupMember 'TestADGroup'| ft name



Рис.2.9 Вывод участников группы

Если в данную группу включены другие группы домена, чтобы вывести полный список членов, в том числе всех вложенных групп, воспользуйтесь параметром **Recursive**.

Get-ADGroupMember ‘server-admins' -recursive| ft name

Чтобы выгрузить список учетных записей, состоящих в определённой группе в CSV файл (для дальнейшего использования в Excel), выполните такую команду:

Get-ADGroupMember ‘server-admins' -recursive| ft samaccountname| Out-File c:\ps\admins.csv

Чтобы добавить в текстовый файл данные учетных записей пользователей в AD, воспользуемся командлетом [Get-ADUser](http://winitpro.ru/index.php/2015/05/21/powershell-get-aduser-poluchenie-dannih-o-polzovatelyax-active-directory/). Например, помимо учетной записи нужно вывести должность и телефон пользователя группы:

Get-ADGroupMember -Identity ’server-admins’ -recursive| foreach { Get-ADUser $\_ -properties title, OfficePhone|Select-Object title, OfficePhone }

Посчитать количество пользователей в группе можно так:

(Get-ADGroupMember -Identity 'domain admins').Count

Оказалось, что в группе «domain admins» у нас состоит 7 учетных записей администраторов.

Чтобы найти список пустых групп в определенном OU, воспользуйтесь такой командой:

Get-ADGroup -Filter \* -Properties Members -searchbase “OU=Moscow,DC=corp,dc=winitpro,DC=ru”  | where {-not $\_.members} | select Name

## 2.6 Управление объектами Active Directory

Обычные задачи, которые вы будете выполнять с помощью службы каталога Microsoft Active Directory системы Windows Server 2003, вовлекут вас в управление такими объектами Active Directory как *пользователи* и *группы.* Большинство компаний создает и реализует проект Active Directory один раз. После развертывания с большинством объектов Active Directory произойдут небольшие изменения. Однако работа с объектами user (пользователь) и объектами group (группа) является исключением из этого правила. По мере того как служащие присоединяются к компании или оставляют ее, администратор тратит время на управление пользователями и группами. Служба Active Directory содержит другие объекты, такие как printer (принтер), computer (компьютер) и shared folder (общие папки), которые также требуют частого администрирования.

В этой главе обсуждаются концепции и процедуры, которые используются для управления объектами Active Directory. В ней обсуждаются типы объектов, которые можно хранить в Active Directory и объясняется, как управлять этими объектами. Показан основной интерфейс, который вы будете использовать для работы с объектами, инструмент Active Directory Users And Computers (Пользователи и компьютеры Active Directory) и некоторые усовершенствования, которые сделаны для этого инструмента в Windows Server 2003.

В службе Active Directory Windows Server 2003 существуют три объекта, которые используются для представления индивидуальных пользователей в каталоге. Два из них, объект *user* (пользователь) и объект *inetOrgPerson,* являются участниками безопасности, которые могут использоваться для назначения доступа к ресурсам вашей сети. Третий объект *contact* (контакт) не является участником безопасности и используется для электронной почты.

Объекты User

Один из наиболее типичных объектов в любой базе данных Active Directory — объект *user.* Объект *user,* подобно любому другому объекту

класса Active Directory, представляет собой совокупность атрибутов. Фактически, он может иметь более 250-ти атрибутов. Этим служба Active Directory Windows Server 2003 сильно отличается от службы каталога Microsoft Windows NT, в которой объекты *user* имеют очень мало атрибутов. Поскольку Active Directory может обеспечить эти дополнительные атрибуты, она полезна именно как служба каталога, а не просто как база данных для хранения опознавательной информации. Active Directory может стать основным местом хранения большей части пользовательской информации в вашей компании. Каталог будет содержать пользовательскую информацию: номера телефона, адреса и организационную информацию. Как только пользователи научаться делать поиск в Active Directory, они смогут найти практически любую информацию о других пользователях.

Когда вы создаете объект *user* , нужно заполнить некоторые из его атрибутов. Как показано на рисунке 2.9, при создании учетной записи пользователя требуется только шесть атрибутов, причем атрибуты *сп* и *sAMAccountName* конфигурируются на основе данных, которые вы вводите при создании учетной записи. Остальные атрибуты, включая идентификатор безопасности (SID), автоматически заполняются системой безопасности.



Рис. 2.9. Обязательные атрибуты учетной записи пользователя, отображаемые инструментом Adsiedit.msc

При создании учетной записи пользователя вы можете назначить значения многим атрибутам объекта *user.* Некоторые из атрибутов нельзя увидеть через интерфейс пользователя (UI), например, атрибут *Assistant* (Помощник). Его можно заполнять, используя скрипт или инструмент

Adsiedit.msc, который обращается к атрибуту напрямую. Можно заполнять скрытые атрибуты в процессе общего импорта информации каталога с использованием утилит командной строки Csvde или Ldifde. Детальную информацию по использованию этих утилит смотрите в Help And Support Center (Центр справки и поддержки). Заполнять невидимые в UI атрибуты необходимо, так как они используются для поиска и изменения объектов. В некоторых случаях скрытый атрибут доступен через диалоговое окно Find (Поиск). Например, в инструменте Active Directory Users And Computers (Пользователи и компьютеры Active Directory) для поиска всех пользователей, которые имеют один и тот же атрибут *Assistant,* используйте вкладку Advanced (Дополнительно) в диалоговом окне Find, чтобы создать запрос, основанный на атрибуте *Assistant* (см. рис. 2.10). В этом окне щелкните на кнопке Field (Поле), выберите User (Пользователь), а затем выберите атрибут, по которому вы хотите сделать поиск. Так можно найти многие скрытые атрибуты.



Рис. 2.10. Поиск учетных записей пользователя по атрибутам, которые не видны в пользовательском интерфейсе

Вы можете просматривать и изменять любой атрибут объекта *user,* используя инструменты Adsiedit.msc или Ldp.exe. Более эффективный способ состоит в использовании скриптов. От этого можно получить значительную выгоду, поскольку Active Directory написана так, чтобы разрешать и поощрять использование скриптов. Информацию об использовании скриптов для автоматизации задач управления службой Active Directory смотрите в центре TechNet Script Center по адресу *http://www.microsoft.com/technet/scriptcenter/default.asp.* TechNet Script Center содержит ресурсы для создания скриптов и типовые сценарии, используемые для расширения административных задач, которые иначе выполняются через консоли управления службой Active Directory. Смотрите также веб-сайт Microsoft Press Online по адресу *http:// www.microsoft.com/mspress/,* где находится бонус-глава по созданию сценариев «Introduction to ADSI Scripting Using VBScript» (Введение в ADSI сценарии на языке VBScript), написанная Майком Малкером (Mike Mulcare).

Большинство административных задач, связанных с обычными пользователями, выполняются при помощи инструмента Active Directory Users And Computers. Чтобы создать с его помощью объект *user,* найдите контейнер, в котором вы хотите создать объект, щелкните правой кнопкой мыши и выберите New>User (Новый>Пользователь). При создании пользователя вы должны ввести Full Name (Полное имя) и User Logon Name (Пользовательское имя для входа в систему). Данные Full Name используются для заполнения атрибута *сп* пользователя, данные User Logon Name становятся значением *sAMAccountName.* После создания пользователя можно обращаться к свойствам объекта для заполнения дополнительных атрибутов пользователя, назначение которых вполне понятно. Наиболее важная вкладка, предназначенная для управления учетной записью пользователя — это вкладка Account (Учетная запись) (см. рис. 2.11). Пользовательские параметры настройки, доступные на вкладке Account, описаны в таблице 2.1.



Рис. 2.11. Вкладка Account для объекта *user*

Табл. 2.1. Свойства учетной записи объекта User Параметры учетной записи

|  |  |
| --- | --- |
| **UserLogonName (Пользовательское имя для входа в систему)** | **Идентифицирует основное имя пользователя (UPN) для данного пользователя.** |
| User Logon Name (Пользовательское имя для входа в систему) (использовалось до Windows 2000) | Идентифицирует имя, применяющееся для входа в более ранние, чем Microsoft Windows 2000, системы, используя формат *domain\username.* |
| Logon Hours (Часы входа в систему) | Устанавливает часы, в которые пользователь может входить в домен. |
| Log On To (Вход на) | Перечисляет компьютеры (используя имена NetBIOS компьютеров), на которые пользователю разрешается вход. |
| Account Is Locked Out (Учетная запись блокирована)  | Указывает на то, что учетная запись была блокирована из-за слишком большого числа неудавшихся попыток входа в систему. |
| Account Options (Опции учетной записи) | Обеспечивает настройку таких параметров, как политики пароля и опознавательные требования. |
| Account Expires (Учетная запись недействительна) | Определяет время окончания срока действия учетной записи. |

Каждый объект в Active Directory должен иметь уникальное имя, но для объекта *user* это простое утверждение может стать довольно сложным, потому что объект *user* фактически имеет несколько возможных имен. В таблице 2.2 перечислены все имена, которые могут быть связаны с именем пользователя *username,* и область действия, в пределах которой это имя должно быть уникальным.

Табл. 2.2. Требования уникальности имени пользователя

|  |  |
| --- | --- |
| **Username (Имя пользователя)** | **Требование уникальности** |
| **First name, initials, last name** (Имя, инициалы, фамилия) | Уникальность не требуется. |
| **Display name** (Отображаемое имя) | Уникальность не требуется. |
| **Full name** (Полное имя) - используется для заполнения атрибута *сп* учетной записи пользователя. По умолчанию полное имя создается из полей First Name, Initials и Last Name диалогового окна New Object-User (Новый объект-пользователь). Его можно изменить, используя Adsiedit.msc | Должно быть уникальным в пределах организационной единицы (OU). |
| **Username (Имя пользователя)** | **Требование уникальности** |
| **User principal name** (Основное имя пользователя). UPN составлено из имени входа в систему и DNS-имени домена или альтернативного UPN, если для леса были сконфигурированы дополнительные UPN-суффиксы. | Должно быть уникальным в пределах леса. |
| **User Logon Name (Pre-Windows 2000)**(Пользовательское имя входа в систему, используемое до Windows 2000) | Должно быть уникальным в пределах домена. |

UPN является очень полезным именем для пользователя. Пользователь может перейти в любой домен леса и войти в систему, используя свое UPN-имя, вместо того чтобы при входе выбирать свой домашний цомен. По умолчанию UPN-суффикс является также DNS-именем для цомена. Вы можете изменять UPN-суффикс, например, использовать различные DNS-имена внутри и вне системы для отображения в интернете. В большинстве случаев SMTP-адрес электронной почты для всех пользователей соответствует внешнему имени DNS. Ваши пользователи, возможно, захотят входить в домен, используя свои адреса SMTP. Вы можете включить эту опцию, добавляя альтернативный UPN-суффикс к лесу и назначая его всем учетным записям пользователя. Чтобы создать дополнительный UPN-суффикс, откройте инструмент Active Directory Domains And Trusts (Домены и доверительные отношения Active Directory), щелкните правой кнопкой мыши на записи Active Directory Domains And Trusts, расположенной в верхней левой области окна, и выберите Properties (Свойства). Напечатайте любой альтернативный UPN-суффикс, который вы желаете использовать.

# 3. Проектная глава

При запуске скрипта poweshell должен быть запущен от имени администратора. При запуске без ключа, будут добаляться и обновляться пользователи. Разработанный скрипт приведен ниже:

 # Функция создания паролей

Function Create-Password {

#Задаем длину пароля 10 символов

[int]$passwordlength = 10

#Количество символов в верхнем регистре

[int]$min\_upper = 3

#Количество символов в нижнем регистре

[int]$min\_lower = 3

#Количество цифр

[int]$min\_number = 3

#Количество символов-разделителей

[int]$min\_symbol = 0

#Соединение различных типов символов

#[int]$min\_misc = ($passwordlength - ($min\_upper + $min\_lower + $min\_number + $min\_symbol))

[int]$min\_misc = ($passwordlength - ($min\_upper + $min\_lower + $min\_number))

#Символы для пароля

$upper = @("A","B","C","D","E","F","G","H","I","J","K","L","M","N","O","P","Q","R","S","T","U","V","W","X","Y ","Z")

$lower = @("a","b","c","d","e","f","g","h","i","j","k","l","m","n","o","p","q","r","s","t","u","v","w","x","y ","z")

$number = @(1,2,3,4,5,6,7,8,9,0)

$symbol = @("!","@","#","%","&","(",")","`"",".","<",">","+","=","-","\_")

# $combine = $upper + $lower + $number + $symbol

$combine = $upper + $lower + $number

$password = @()

#Добавление больших символов в пароль

1..$min\_upper | % {$password += Get-Random $upper}

# Добавление маленьких символов в пароль

1..$min\_lower | % {$password += Get-Random $lower}

#Добавление цифр в пароль

1..$min\_number | % {$password += Get-Random $number}

#Add symbols into password

# 1..$min\_symbol | % {$password += Get-Random $symbol}

#Fill out the rest of the password length

1..$min\_misc | % {$password += Get-Random $combine}

#Randomize password

Get-Random $password -count $passwordlength | % {[string]$randompassword += $\_}

Return $randompassword

}

#Функция создания пользователей из файла

Import-Csv -Path "C:\scripts\Tablitsa.csv" | ForEach-Object {$\_.accountpassword=create-password; $p=ConvertTo-SecureString $\_.accountpassword -asPlainText -force; New-ADUser -Name $\_.name -AccountPassword $p -logonworkstations $\_.logonworkstations} | ConvertTo-Html name, samaccountname, accountpassword | out-file "c:\usr.html"

# 4**. Технико-экономическое обоснование и расчет экономической эффективности предлагаемого проекта**

Совокупная стоимость владения (ССВ) является суммой затрат на инвестиционную часть проекта внедрения организационно-технических средств защиты информации, текущих расходов в течение срока жизненного цикла проекта и платежей в виде налогов и других отчислений, связанных с реализацией проекта.

ССВ = P АПК + P ТЕК + P Пл (1)

где P АПК - совокупная стоимость затрат на разработку и реализацию организационно-технической системы защиты информации; PТЕК - совокупная стоимость текущих расходов в течение срока жизненного цикла проекта;PПл - платежи в виде налогов и других отчислений, связанных с реализацией проекта.

## 4.1 Расчет затрат на разработку и реализацию проекта

Одним из этапов оценки экономической эффективности проекта является оценка затрат на приобретение оборудования, разработку или приобретение программ, средств информационного обеспечения, подготовку помещений, движение и обучение персонала.

Затраты по заработной плате персонала Рзп при реализации комплексной системы защиты информации рассчитываются по следующей формуле (2)

 (2)

Где:

Зl - среднемесячная основная заработная плата работника l, руб.;

Tl - среднемесячные трудовые затраты работника на выполнение исследуемой операции, чел./час;

 - ставка страховых взносов (=34%);

 - дополнительная заработная плата (премия);

Q - среднее количество рабочих часов в месяце, час.

Перечень сотрудников ОАО «Кузнецкий металлургический комбинат», занятых в обеспечении информационной безопасности представлен в таблице 3.1.

Таблица 3.1 Сотрудники, обеспечивающие разработку и внедрение

|  |  |  |
| --- | --- | --- |
| Название должности | Количество штатных единиц | Заработная плата, руб |
| Разработчик | 1 | 40000 |
| Системный аминистратор | 1 | 40000 |

В нашем случае затраты по заработной плате персонала, работающего с составлением различных положений и регламентов исходя из трудоемкости ведения данных работ составят:

Рзп= (40000\*58+40000\*116)\*1,34/176 = 49032 руб.

Затраты на настройку программных средств защиты и затраты на проведение инженерно-технических работ по установке средств защиты исходя из трудоемкости ведения данных работ составят:

РИТ-персонала = (10000\*60\*1,34)/176= 4569 руб.

Руст = = 46410 руб.

Обучение персонала работе с предложенной комплексной системой защиты информации не требуется.

Инвестиционный план проекта сведен в таблицу 5.2.

Таблица 3.2 Инвестиционный план проекта

|  |  |
| --- | --- |
| Статьи расходов | Денежные затраты в тыс. руб. |
|  | 0 | 1 | 2 | 3 |
| Расходы, связанные с организационно-правовым направлением защиты |
| Затраты на разработку положений и регламентов | 49 |  |  |  |
| Расходы, связанные с программно-аппаратным направлением защиты |
| Затраты на разработку или/и приобретение программных средств | 99,4 | 0,9 | 0,9 | 0,9 |
| Затраты, связанные с обслуживанием и настройкой программных средств защиты | 4,6 |  |  |  |
| Расходы, связанные с инженерно-техническим направлением защиты |
| Затраты на приобретение средств инженерно-технической защиты | 51,5 |  |  |  |
| Затраты на установку и настройку инженерно-технических средств защиты | 46,4 |  |  |  |
| Итого за проект | 250,9 | 251,8 | 252,7 | 253,6 |

## 4.2 Расчет текущих расходов в течение срока жизненного цикла проекта

Данный этап экономического обоснования проектных решений в области информационной безопасности связан с определением состава и объёма затрат в периоды жизненного цикла ИТ, соответствующие процессам эксплуатации спроектированного комплекса - текущих затрат Ртек за каждый из периодов m.

В рамках данной работы предусматриваются следующие текущие затраты:

* Затраты на оплату труда управленческого персонала;
* Затраты на связь, литературу, канцтовары, связанные с поддержкой процессов информационной безопасности;
* Затраты, связанные с обслуживанием и настройкой программных средств защиты;
* Затраты на оплату расхода электроэнергии на инженерно-технический комплекс;
* Затраты на проведение инженерно-технических работ по установлению средств защиты.

Затраты, связанные с обслуживанием и настройкой программных средств защиты рассчитываются по формуле (3):

 (3)

Где  - расходы по оплате труда ИТ-персонала, занятого настройкой и обслуживанием программных средств защиты, которые рассчитываются по формуле (2), а  - материальные затраты, связанные с обслуживанием и восстановлением системы исходя из трудоемкости ведения данных работ.

В нашем случае  = (10000\*60\*1,34)/176= 4569 руб.

Затраты на оплату расхода электроэнергии на комплекс приобретенных средств рассчитываются по формуле (4):

 (4)

Где Мпотр - максимальная потребляемая мощность оборудования g, Кисп - коэффициент использования оборудования g по мощности, Сэл - стоимость 1 кВт/час электроэнергии, Тg - время работы оборудования g.

Максимальная потребляемая мощность оборудования, задействованного в программно-аппаратной и инженерно-технической системах защиты информации представлена в таблице 3.3.

Таблица 3.3 Установленная мощность оборудования, задействованного в программно-аппаратной системе защиты информации

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Установленная мощность, Вт. | Количество, шт. | Итого, Вт. |
| Время работы - 24 часа, 365 дней в году |
| Cisco 1921 Integrated Services Router | 13 | 1 | 13 |
| ППК «Сигнал-20» | 18 | 1 | 18 |
| Магнитоконтактный охранный извещатель ИО-102-6 (СМК-6) | 10 | 6 | 60 |
| Извещатель охранный звуковой GBD-2 | 12 | 6 | 72 |
| Источник бесперебойного электропитания «Скат-1200» | 34 | 2 | 68 |
| Оповещатель совмещенный «Свирель-2» | 14 | 1 | 14 |
| Монитор Ai-ML247N | 42 | 1 | 42 |
| Миниатюрная видеокамера KT&C KPC-190SP4 | 18 | 5 | 90 |
| Видеорегистратор «PVDR-1654» | 3 | 1 | 3 |
| Итого: |  |  | 380 |

Затраты на оплату расхода электроэнергии на инженерно-технический комплекс в год составят

Эн = 0,380\*1\*5,2\*24\*365 = 17310 руб.

Текущие затраты на обеспечение процессов ИБ сведены в таблицу 3.4.

## 4.3 Расчет платежей в виде налогов и экономии налогов на прибыль за счет амортизации

К налогам и другим отчислениям относится налог на имущество организации. Сумма платежей за период B рассчитывается по формуле (5):

 (5)

где SостBk - остаточная стоимость k -ого вида имущественной номенклатуры на начало периода, L - ставка налога на имущество (L=2,2%), N - количество имущественных номенклатур; Ln- ставка налога на прибыль, А- амортизационные отчисления. Остаточная стоимость имущественной номенклатуры на начало периода i рассчитывается по формуле (7):

 (6)

где На - норма амортизации.

Норма амортизации определяется по формуле (8):

 (7)

где: Sлик - ликвидационная стоимость имущественной номенклатуры; Тэ срок эксплуатации.

Норма амортизации для имущества ОАО «Кузнецкий металлургический комбинат» представлена в таблице 3.5.

Пользуясь формулой (5) была определена сумма платежей за три года. Результаты приведены в таблице 3.7.

Таблица 3.4 Сумма платежей

|  |  |
| --- | --- |
| Наименование имущества | Сумма отчислений, тыс. руб. |
|  | 1 | 2 | 3 |  |
| Все виды | 1,25 | 1,06 | 0,94 | 3,25 |
| Итого | 3,25 |

Используя формулу (1) рассчитаем совокупную стоимость владения

ССВ=253,6+3\*(22,6+3,4+17,3)+3,25=386,85 тыс. руб.

# Заключение

Windows PowerShell — новая командная строка и язык сценариев от компании Microsoft. PowerShell является компонентом Windows Server 2008 (надо только выбрать его в Server Manager) и доступна для загрузки со странички www.microsoft.com/powershell для Windows XP, Windows Server 2003 и Windows Vista.

Итак, команды PowerShell называются командлетами (cmdlet) и состоят из глагола (например, get, set, new, remove, move, connect) и существительного в единственном числе, описывающего объект действия. Между ними ставится дефис. Получается что-то вроде: get-process, stop-service и т.п.

Команды, как правило, связываются конвейером, обозначаемым вертикальной чертой (|). Этот знак означает, что вся коллекция объектов из предыдущей команды передается на вход следующей.

Такая объектная ориентированность очень удобна, поскольку позволяет легко оперировать объектами и связывать команды вместе. В этой статье мы расскажем, как подобный подход облегчает управление корпоративной директорией на базе Active Directory.

# Список литературы

1. Payette B. Windows PowerShell in Action. — Manning Publications Co, 2007.

2. Holmes L. Windows PowerShell Cookbook. — O'Reilly, 2007.

3. Wilson E. Windows PowerShell Step-by-Step. — Microsoft Press, 2007.

4. Watt A. Professional Windows PowerShell (Programmer to Programmer). —

Wrox Press, 2007.

5. Kopczynski T. Windows PowerShell Unleashed. — SAMS Publishing, 2007.

6. Jones D., Hicks J. Windows PowerShell: TFM. — Sapien, 2006.

7. Cookey-Gam J., Keane B., Rosen J., Runyon J., Stidley J. Professional Windows

PowerShell for Exchange Server 2007 Service Pack 1 (Programmer to

Programmer). — Wrox Press, 2008.

8. Koch F. Windows PowerShell. — (Электронная версия книги доступна для

бесплатного скачивания по адресу http://blogs.technet.com/chitprode/archive/2007/05/10/english-version-of-windows-powershell-coursebook-available-for-download.aspx).

9. Koch F. Administrative tasks using PowerShell. — (Электронная версия книги

доступна для бесплатного скачивания по адресу http://blogs.technet.com/

chitpro-de/archive/2008/02/28/fre