Windows

- Connect Office 365 Exchange Online Services to PowerShell
- Автоматическая репликация областей отказоустойчивого DHCP сервера Windows Server
- Полный список переменных сред в Windows 10
- Управление ролями и функциями Windows Server с помощью PowerShell
- Сертификат для Linux в центре сертификации Active Directory
- Репликация Active Directory
- Подпись файлов signtool.exe

Connect Office 365 Exchange Online Services to PowerShell

How to Connect Office 365 Exchange Online Services to PowerShell

The default method for managing Microsoft Office 365 with all included applications is by using a web browser and connecting to the web interface of Office 365 Admin Center and Exchange Admin Center. This standard method has an intuitive graphical interface, but sometimes the capabilities of the graphical user interface are not enough. For example, when you need to perform similar actions with tens or hundreds of user accounts, it is better to use the CLI (command line interface) rather than the GUI (graphical user interface). Moreover, some actions that can be done with Exchange Online cmdlets are not available in the Admin Center.

Microsoft provides PowerShell to manage the products in the command line interface. Many administrators have the habit of managing Microsoft Exchange Server via PowerShell. However, standard commands of the Exchange PowerShell module that work in the local environment with a standalone instance of Microsoft Exchange Server installed on a physical server or a virtual machine cannot be used with Exchange 365 running in the cloud. This is because standard PowerShell commands cannot connect to cloud services such as Azure and Office 365 backup, including Exchange Online. For this reason, you will need to install special PowerShell modules that allow you to connect to Office 365. The details of this process are covered in today's blog post that explains multiple methods of how to connect to Exchange Online PowerShell and consists of the following parts:

- Requirements
- The Working Principle
- Manual Configuration
- Automated Configuration
- The Alternative Method

NAKIVO Backup & Replication offers improved backup capabilities by adding Microsoft Office 365 backup to its feature set. Start the Free Trial of our product, discover how NAKIVO Backup & Replication works in your environment, and receive an Amazon eGift card for your feedback.

Requirements

There are some requirements that must be met to connect to Exchange Online PowerShell.

- You should use PowerShell on Windows 7 SP1, or newer desktop Windows versions and Windows Server 2008 R2 SP1, or newer server Windows versions. Be aware that you need to install .NET Framework 4.5 or later in addition to installing an updated version of Windows Management Framework 3.0, 4.0, or 5.1.
- An internet connection is required. TCP port 80 must be opened to connect from your local machine to the destination host.
- Access to Exchange Online PowerShell must be enabled for the current user (by default such access is enabled for administrators).

You can manually enable access to connect to Exchange Online PowerShell for the particular user with the command:

Set-User -Identity user@domain.com -RemotePowerShellEnabled $true

Working Principle

You can connect to Exchange Online PowerShell, but this process is more complicated than using PowerShell for managing a local Exchange Server. However, you can use the built-in PowerShell console to manage remote cloud infrastructures. In this case, the console is called *remote PowerShell* or *PowerShell Remoting*. The process of initiating a remote PowerShell session for Office 365 and Exchange Online is slightly different. You should download and install special components before you can open a remote Office 365 session. Fortunately, the cmdlets required to initiate a remote Exchange Online PowerShell are downloaded automatically when you create a remote PowerShell session. Different sets of PowerShell cmdlets are used to manage Microsoft Office 365 and Microsoft Exchange Online.

The main reasoning behind connecting to Microsoft Exchange Online in PowerShell entails the following:

- Creating a remote session to Exchange Online in PowerShell opened on your local machine.
- Providing connection settings, passing authentication.
- Importing PowerShell cmdlets that are needed to manage Exchange Online remotely.

In today’s blog post, we will run PowerShell cmdlets on Windows 10.

Manual Configuration

Let’s review the manual method first, to understand the configuration principle better.

1. Open Windows PowerShell. You can do this with at least two methods.

1a. Click **Start**, type **cmd**, right click the **Command Prompt** item and select *Run as Administrator* in the context menu.

1b. Go to **Start > Windows PowerShell**. Right click Windows PowerShell, and hit *Run as Administrator* to make sure that you can run PowerShell commands without restrictions.

2. Enable running scripts (it is better to run this command in the beginning of preparing PowerShell to manage Exchange Online and Office 365), otherwise you will get the error in future when running the **Import-PSSession** command:

Import-PSSession : Files cannot be loaded since running scripts has been disabled on this system. Provide a valid certificate with which to sign the files.

In order to execute scripts, the execution policy must be set to *RemoteSigned*.

Set-ExecutionPolicy RemoteSigned

Press **Y** to confirm changing the policy if prompted. You can also use the **Set-ExecutionPolicy Unrestricted** command to use the *Unrestricted* policy. By default, the execution policy mode is *Restricted*.

Change the execution policy before you connect to Exchange Online PowerShell

3. Run the command in PowerShell to get credentials and enter your administrator login/password in the popup window to access Exchange Online. The user must have global administrative permissions in Office 365.

$Credential=Get-Credential

How to connect to Office 365 PowerShell – saving credentials as the variable

The entered credentials will be saved in the variable and used in the next command as *$Credential*.

4. You have to create a remote PowerShell session with the *New-PSSession* cmdlet and running the following command:

$Session = New-PSSession -ConfigurationName Microsoft.Exchange -ConnectionUri https://outlook.office365.com/powershell-liveid/ -Credential $Credential -Authentication Basic -AllowRedirection

Notice that in this command, the target URL of the Exchange Online server running in the cloud that must accept the request is set. After running the command, Microsoft Office 365 cloud servers will provide you access to the appropriate Exchange Online virtual server associated with your account.

How to connect to Exchange Online PowerShell – creating a new remote session

5. Exchange Online PowerShell cmdlets must be imported to the current session with the command:

Import-PSSession $Session

You can see the progress bar while receiving the commands.

How to connect to Exchange Online PowerShell – importing Exchange online cmdlets to the current s

After successfully running the command, you will see the following message.

How to connect to Exchange Online PowerShell – Exchange Online cmdlets have been imported succ

Note: If you use the MFA for your account, the standard cmdlets explained above will not work. If you want to connect Exchange Online in PowerShell using MFA, run the following command:

Connect-EXOPSSession -UserPrincipalName YOUR_UPN

Where YOUR_UPN (user principle name) is the name of the Office 365 account you are using.

You may need to install Microsoft's *Exchange Online Remote PowerShell Module.* Be aware that when using this module, the session ends after one hour, which may be inconvenient for running long scripts. Consider using *Trusted IP addresses* (i.e. the IP addresses of your organization) to bypass MFA when connecting from the network of your company to Exchange Online PowerShell.

MFA (Multi-Factor-Authentication) is the advanced method of authentication that adds a second layer of security. After entering a password, the confirmation code is sent to the user's cell phone and the user must enter the confirmation code to verify the account to get access to Office 365 cloud services.

6. Once you have connected to Office 365 and Exchange Online, you can manage your Office 365 cloud environment. Let's verify that we have connected to Exchange Online correctly and list the mailboxes of users, for example.

Get-Mailbox

Exchange Online PowerShell – checking mailboxes of users

You can list all cmdlets available for Exchange Online PowerShell with the following command:

Get-Command -Module tmp*

The names of Exchange Online PowerShell cmdlets are not converted.

7. When you end your work with Exchange 365, disconnect the session. This is the recommended practice.

Remove-PSSession $Session

Unfortunately, no messages are displayed after executing this command. You can check whether the session is disconnected by running the **Get-MailBox** command. If the session is disconnected, you will get the error explaining that you cannot run Exchange Online cmdlets after disconnecting.

Why should you disconnect the session? Well, simply because the number of active concurrent sessions that can be opened simultaneously is limited to three. If you open three Exchange Online PowerShell sessions at once and don't disconnect any of them when not in use, you will need to wait until one of these sessions expires before you can connect to Exchange Online PowerShell again from a new PowerShell console.

Automated Configuration

Now that you know the principle of how to connect to Exchange Online PowerShell manually, you can use the automated method. The advantage of this method is the lower number of commands you should enter.

1. Download the script from the Microsoft's web site. The name of the script file is *ConnectExchangeOnlinePowerShell.ps1* in this case.

2. Go to the directory where the script is located; in our example, the script is saved to *C:\temp_win*.

3. Before running the script, edit the script execution policy (similarly to what is shown in the first method), otherwise you will get the error:

The file C:\temp_win\ConnectExchangeOnlinePowerShell.ps1 is not digitally signed. You cannot run this script on the current system.

You can apply the *Bypass* execution policy to avoid this issue:

Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass

Type **Y** to confirm the execution policy change.

Connect to Exchange Online PowerShell by using a PowerShell script

4. After that, you can run the script. If you don't use MFA, run the script without additional arguments:

.\ConnectExchangeOnlinePowerShell.ps1

If MFA (Multi-Factor Authentication) is used in your Office 365 environment, try the command:

.\ConnectExchangeOnlinePowerShell.ps1 -MFA

5. Now that you have successfully connected to Exchange Online, you can manage your user accounts, their mailboxes, etc. For example, you can list the mailboxes of your users:

Get-Mailbox

Exchange Online PowerShell is now connected – checking mailboxes

This script can be used to schedule and automate tasks. For example, you can connect to Exchange Online without entering credentials in the interactive window as shown above. You can enter your login and password in the command line as command options when executing the script:

./ConnectExchangeOnlinePowerShell.ps1 -UserName admin@your_domain.com -Password your_password

Keep in mind that entering passwords as plain text in the command line may be not secure.

6. When you finish to work with Exchange Online in PowerShell, don’t forget to end the session:

./ConnectExchangeOnlinePowerShell.ps1 -Disconnect

Alternative Method

Let’s consider one more method that can be used to connect to Exchange Online PowerShell. This method can be considered as a modification of the first method.

1. Create a new profile for PowerShell with the function:

New-item -type file -force $profile

2. Edit the profile configuration file in the text editor to add the function titled *Connect-EXOnline*:

notepad $profile

3. Add the following content to the PowerShell profile configuration file and change **username@domain.com** to your account name, then save the text file.

Function Connect-EXOnline

{

$credentials = Get-Credential -Credential username@domain.com

Write-Output “Getting the Exchange Online cmdlets”

$Session = New-PSSession -ConnectionUri https://outlook.office365.com/powershell-liveid/ `

-ConfigurationName Microsoft.Exchange -Credential $credentials `

-Authentication Basic -AllowRedirection

Import-PSSession $Session

}

4. Close the current PowerShell window and open a new PowerShell window as Administrator. Run the command to connect to Exchange Online PowerShell:

Connect-ExOnline

Enter your password in the popup window.

PowerShell connect to Exchange Online

5. When you have finished working with Exchange Online PowerShell, end the session with the command:

Get-PSSession | Remove-PSSession

Автоматическая репликация областей отказоустойчивого DHCP сервера Windows Server

> Источник http://pyatilistnik.org/oshibki-20291-i-20292-na-dhcp-v-windows-server/

Как я написал выше, Microsoft до сих пор не решило проблему автоматической репликации данных, о зарезервированных IP-адресах, даже в Windows Server 2019. Тут у вас два варианта, первый самый топорный, это ручной запуск на сервере, где были изменены настройки. Для этого запустите PowerShell и введите команду:

```
Invoke-DhcpServerv4FailoverReplication -ComputerName dc01.root.pyatilistnik.org
```

- dc01.root.pyatilistnik.org - сервер с которого мы будем реплицировать настройки (**DHCP сервер-партнер**)

В этом примере реплицируются **все области** отработки отказа службы DHCP-сервера, работающей на компьютере с именем dc01.root.pyatilistnik.org, в одну или несколько соответствующих партнерских служб DHCP-сервера на основе одного или нескольких отношений отработки отказа, в которых включены службы DHCP-сервера. Обратите внимание, что у вас появится запрос на подтверждение данного действия. Если хотите его пропускать, то необходимо добавить ключ **-Force**.

```
Invoke-DhcpServerv4FailoverReplication -ComputerName dc01.root.pyatilistnik.org -Force
```

Команда Invoke-DhcpServerv4FailoverReplication -ComputerName dc01.root.pyatilistnik.org

Если вам необходимо произвести репликацию настроек отработки отказа относительно одной группы, то команда будет выглядеть вот так:

```
Invoke-DhcpServerv4FailoverReplication -ComputerName dc01.root.pyatilistnik.org -Name dc01.root.pyatilistnik.org-svt2019s01
```

В этом примере реплицируется конфигурация всех областей, которые являются частью отношения отработки отказа с именем **dc01.root.pyatilistnik.org-svt2019s01** в службе DHCP-сервера, работающей на компьютере с именем **dc01.root.pyatilistnik.org**, в службу DHCP-сервера партнера.

Репликация отработки отказа через PowerShell

Предположим, что вам необходимо выполнить репликацию отработки отказа, только для определенных областей, для этого есть параметр -ScopeId, вот пример для моих областей:

```
Invoke-DhcpServerv4FailoverReplication -ComputerName dc01.root.pyatilistnik.org -ScopeId 192.168.31.0,192.168.32.0 -Force
```

Зная теперь основные команды, вы можете создать для себя автоматический сценарий при котором у вас будет происходить репликация. Тут все просто на поможет сценарий PowerShell и планировщик заданий, который будет срабатывать на определенное событие в журналах Windows.

Хочу отметить, что я и сама Microsoft настоятельно рекомендует, всем администраторам делать изменение настроек исключительно на одном из серверов партнеров, а уже потом реплицировать на второй, так можно гарантировать, что все реплики будут корректны и минимизировано количество потерь ваших изменений

Алгоритм действий:

- Создаете в Active Directory новую учетную запись, например dhcp-replication и **делаете ее администратором на обоих DHCP серверах**, кто входит в группу репликации. Как дать права на DHCP читайте по ссылке.

Создание учетной записи для репликации областей отработки отказа

- Создаем небольшой скрипт dhcp-replication.ps1. Для этого вам просто нужно открыть PowerShell ISE и ввести там команду:

Invoke-DhcpServerv4FailoverReplication -ComputerName dc01.root.pyatilistnik.org -Force#dc01.root.pyatilistnik.org - сервер с которого мы будем реплицировать настройки (**DHCP сервер-партнер**)

Далее вам просто нужно сохранить его в формате ps1.

Создание PowerShell скрипта для репликации DHCP

- Создаем задание в планировщике Windows, которое по событию будет запускать ваш скрипт. Почему именно по событию, все просто, чтобы уменьшить количество бесполезного трафика и нагрузку на ваш сервер.

Когда вы производите новое резервирование IP-адреса на вашем DHCP сервере, у вас генерируется событие ID 106.

ID 106: Резервирование: [[192.168.31.109]] для IPv4 настроено в области [[192.168.31.0]192.168.31.0] через ROOT\Администратор.

Генерируется оно журналом **Microsoft-Windows-DHCP Server Events**.

Событие с кодом ID 106

И так, откройте планировщик событий, самый быстрый способ, это в окне "**Выполнить**", ввести **taskschd.msc**.

как открыть планировщик заданий

Щелкаем правым кликом и создаем новую, простую задачу.

Создание задания по репликации в DHCP

На первом шаге, вам необходимо придумать имя для вашей задачи, можно написать, что угодно, главное, чтобы вам было понятно.

Заполнение поля имени задачи в планировщике

Выбираем пункт "**При занесении в журнал указанного события**".

При занесении в журнал указанного события

Далее заполняем три пункта:

- Журнал - Microsoft-Windows-DHCP Server Events/Работает
- DHCP-Server
- Код события 106

> 1. создание задания в планировщике Windows по репликации областей DHCP

Оставляем пункт "**Запустить программу**".

Запуск программы через планировщик Windows

Далее в поле:

- Программа или сценарий, вы указываете powershell.exe
- В поле "Добавить аргументы" введите текст **-WindowStyle Hidden -File "C:\Scripts\dhcp-replication.ps1"**. Где **WindowStyle** - это указание не показывать диалоговое окно, а атрибут -**File** указывает путь до вашего скрипта PowerShell. Мы такое уже разбирали в методах запуска скрипта PowerShell.

Заполняем аргументы для запуска скрипта PowerShell

Смотрим сводную информацию и закрываем мастер настройки простого задания.

Завершение создания задания в планировщике

В результате ваше новое задание появится в списке.

Список заданий в планировщике

Далее нам необходимо открыть свойства вашего задания и в параметрах безопасности выбрать созданного ранее пользователя, от имени которого будет запускаться задание, далее выставить опцию "**Выполнить для всех пользователей**" и поставить галку "**Выполнить с наивысшими правами**".

Все теперь ждем появления события ID 106, вы его сами можете вызвать, путем резервирования IP-адреса. Как видим у меня успешно отработало мое задание и все резервированные IP-адреса на текущем сервере, благополучно отреплецировались на DHCP-партнера.

С событием установки резервирования (ID 106) мы разобрались, но еще есть и обратная операция, это удаление резервирования IP-адреса за определенным компьютером, и тут генерируется событие **ID 107**.

ID 107: Резервирование: [[192.168.31.118]] для IPv4 удалено в области [[192.168.31.0]192.168.31.0] через ROOT\Администратор.

- Журнал Microsoft-Windows-DHCP Server Events/Работает
- Источник DHCP-Server

В итоге вам нужно создать такое же задание, как и в случае с ID 106, но сделать для ID 107 некоторые изменения. Теперь при включении и выключении аренды IP адреса, все области отработки будут автоматически согласованы и ошибок с ID 202091 и ID 20292 не должно быть.

Полный список переменных сред в Windows 10

VARIABLE	WINDOWS 10
%ALLUSERSPROFILE%	C:\ProgramData
%APPDATA%	C:\Users\{имя пользователя}\AppData\Roaming
%COMMONPROGRAMFILES%	C:\Program Files\Common Files
%COMMONPROGRAMFILES(x86)%	C:\Program Files (x86)\Common Files
%CommonProgramW6432%	C:\Program Files\Common Files
%COMSPEC%	C:\Windows\System32\cmd.exe
%HOMEDRIVE%	C:\
%HOMEPATH%	C:\Users\{имя пользователя}
%LOCALAPPDATA%	C:\Users\{имя пользователя}\AppData\Local
%LOGONSERVER%	\\{domain_logon_server}
%PATH%	C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem
%PathExt%	.com;.exe;.bat;.cmd;.vbs;.vbe;.js;.jse;.wsf;.wsh;.msc
%PROGRAMDATA%	C:\ProgramData
%PROGRAMFILES%	C:\Program Files
%ProgramW6432%	C:\Program Files
%PROGRAMFILES(X86)%	C:\Program Files (x86)
%PROMPT%	PG
%SystemDrive%	C:
%SystemRoot%	C:\Windows
%TEMP%	C:\Users\{имя пользователя}\AppData\Local\Temp
%TMP%	C:\Users\{имя пользователя}\AppData\Local\Temp
%USERDOMAIN%	Пользовательский домен, связанный с текущим пользователем.
%USERDOMAIN_ROAMINGPROFILE%	Пользовательский домен, связанный с перемещаемым профилем.

VARIABLE	WINDOWS 10
%USERNAME%	{имя пользователя}
%USERPROFILE%	C:\Users\{имя пользователя}
%WINDIR%	C:\Windows
%PUBLIC%	C:\Users\Public
%PSModulePath%	%SystemRoot%\system32\WindowsPowerShell\v1.0\Modules\
%OneDrive%	C:\Users\{имя пользователя}\OneDrive
%DriverData%	C:\Windows\System32\Drivers\DriverData
%CD%	Выводит текущий путь к каталогу. (Командная строка.)
%CMDCMDLINE%	Выводит командную строку, используемую для запуска текущего сеанса командной строки. (Командная строка.)
%CMDEXTVERSION%	Выводит количество текущих расширений командного процессора. (Командная строка.
%COMPUTERNAME%	Выводит имя системы.
%DATE%	Выводит текущую дату. (Командная строка.)
%TIME%	Время выхода. (Командная строка.)
%ERRORLEVEL%	Выводит число определяющих статус выхода предыдущей команды. (Командная строка.)
%PROCESSOR_IDENTIFIER%	Идентификатор процессора
%PROCESSOR_LEVEL%	Outputs processor level.
%PROCESSOR_REVISION%	Вывод ревизии процессора.
%NUMBER_OF_PROCESSORS%	Выводит количество физических и виртуальных ядер.
%RANDOM%	Выводит случайное число от 0 до 32767.
%OS%	Windows_NT

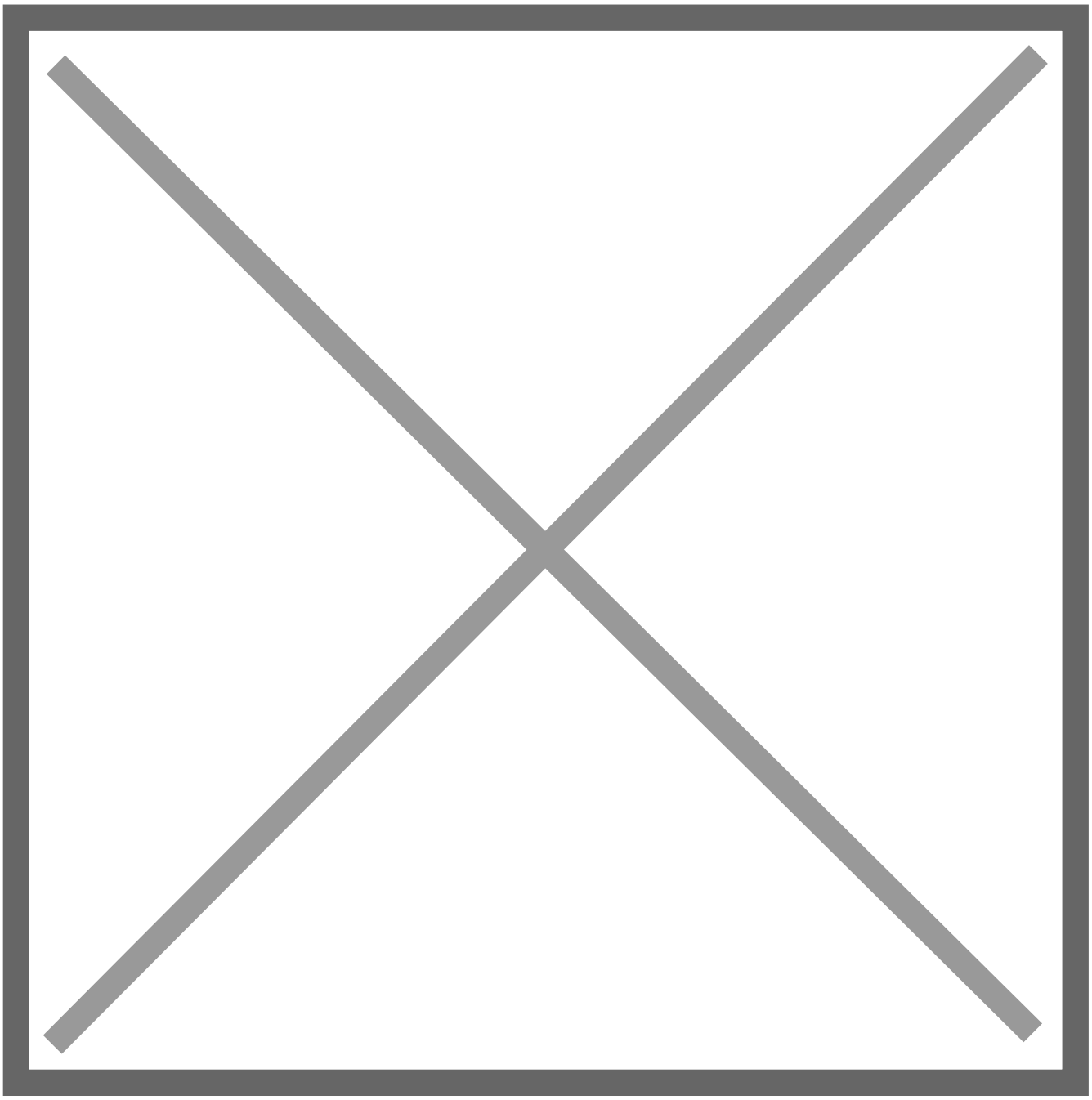
> Помните, что некоторые из упомянутых переменных не зависят от местоположения, в том числе **% COMPUTERNAME%, %PATHEXT%, %PROMPT%, %USERDOMAIN%, %USERNAME%.**

Управление ролями и функциями Windows Server с помощью PowerShell

Как через PowerShell вывести список всех установленных ролей и функций Windows Server

Используйте командлет Get-WindowsFeature, чтобы отобразить список всех доступных ролей и компонентов Windows Server. Если вы запустите его без параметров, вы увидите информацию обо всех компонентах Windows Server.

1	`Get-WindowsFeature`



Отображаются имя компонента (**Display Name**), его системное имя (**Name**) и состояние (**Install State**: **Installed**, **Available** или **Removed**). Список ролей и функций выглядит как дерево с вложенными ролями, подобное тому, которое вы видите при установке ролей в графическом интерфейсе Server Manager. Чтобы установить и удалить любые роли или компоненты с помощью PowerShell, вы должны знать их системные имена, указанные в столбце «Имя».

Подсказка: Если роль или функция удалены, это означает, что её установочные файлы удалены из системного компонента sote (для уменьшения размера папки WinSxS), и вы не сможете установить роль без доступа в Интернет или установочного ISO диска с Windows Server.

Вы можете удалить роли или компоненты из хранилища следующим образом:

1	`Uninstall-WindowsFeature -Name DHCP -Remove`

Чтобы установить удалённую роль DHCP, используйте этот командлет (вам понадобится прямой доступ в Интернет):

1	`Install-WindowsFeature DHCP`

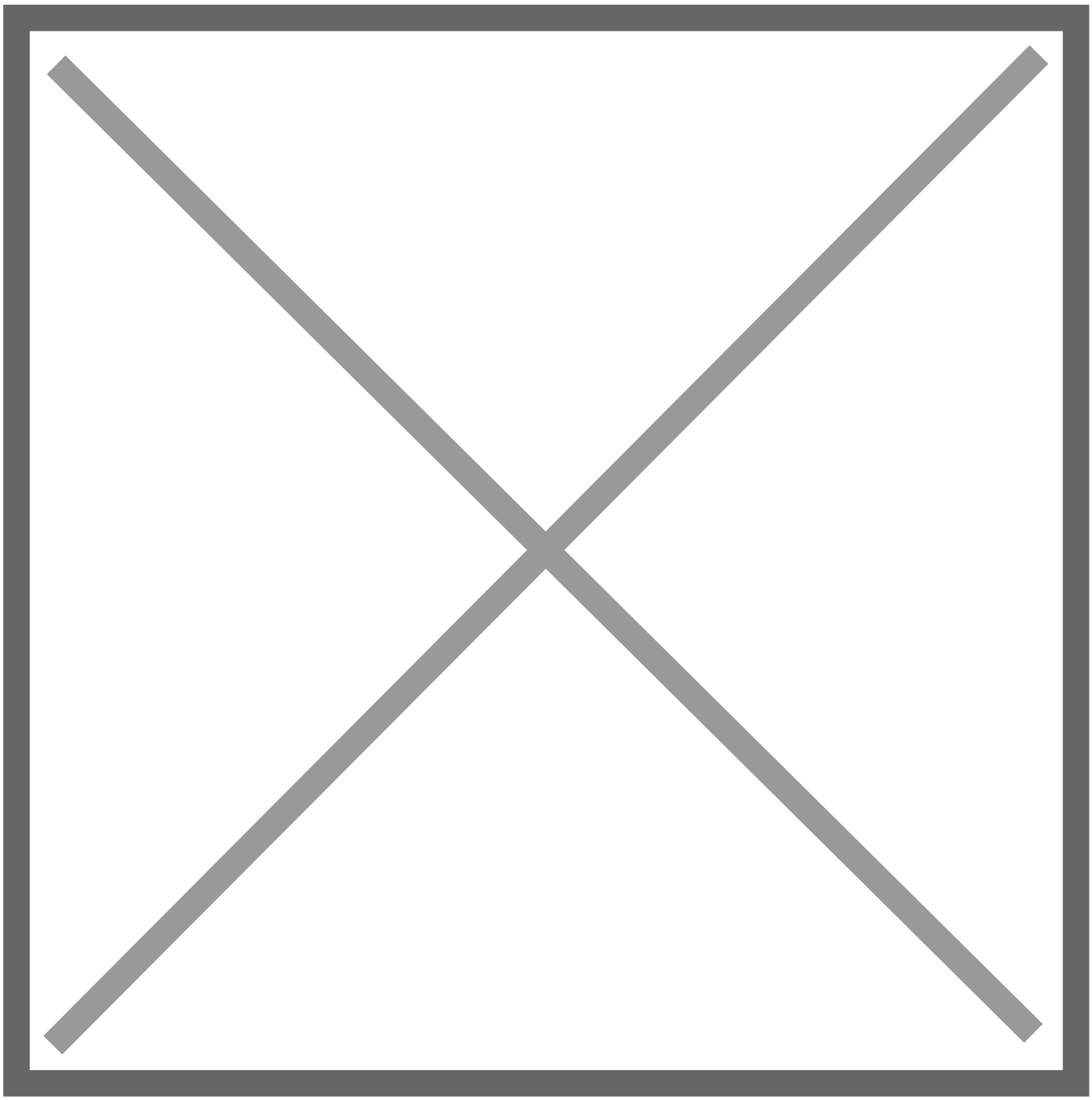
Или вы можете восстановить двоичные файлы компонентов из ISO-образа Windows Server:

1	`Install-WindowsFeature DHCP -Source E:\sources\sxs`

Вы можете перечислить установленные функции сервера:

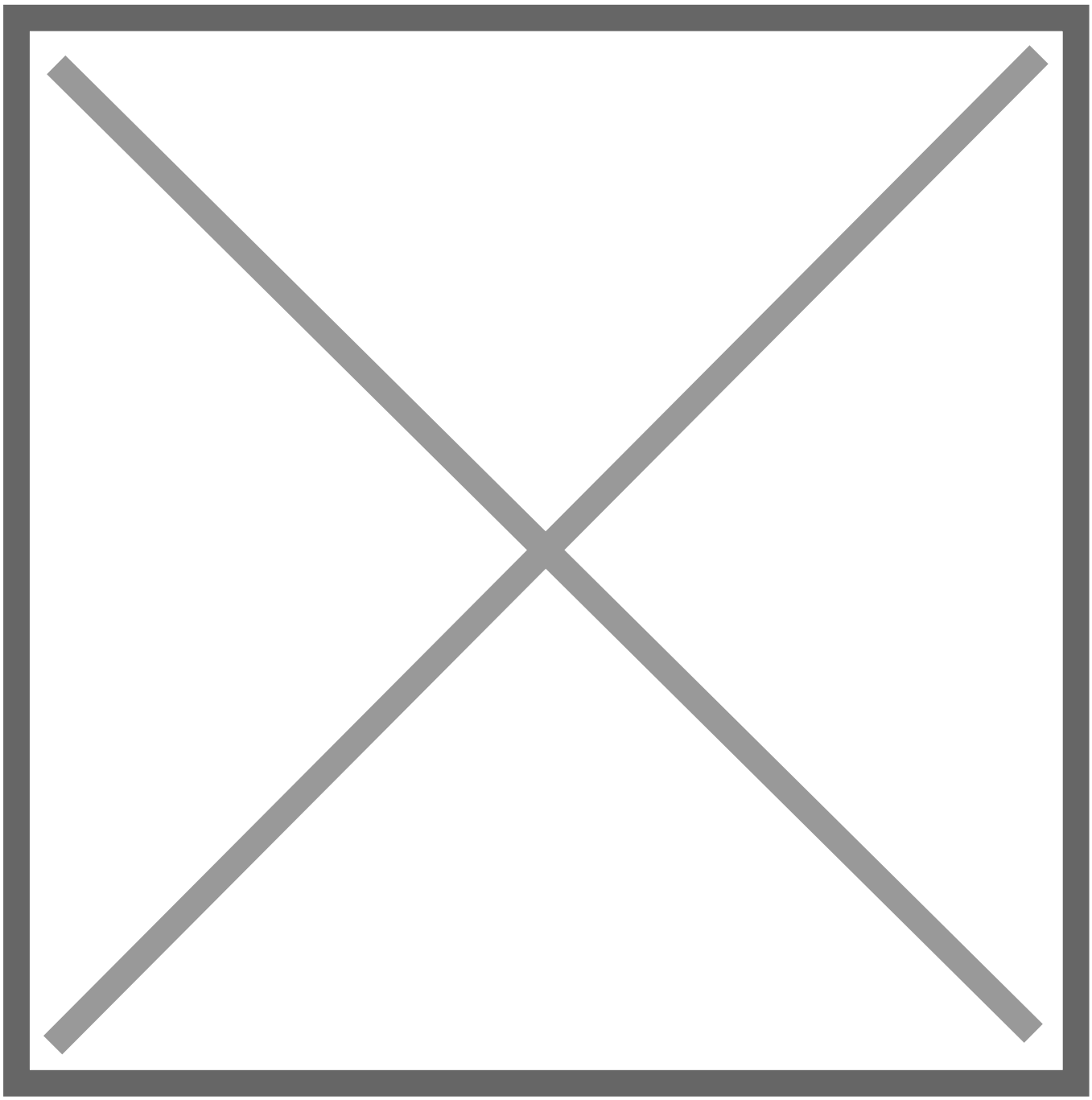
1	`Get-WindowsFeature | Where-Object {$_. installstate -` `eq` `"installed"` `} | ft Name,Installstate`

Судя по приведённому ниже снимку экрана, этот сервер используется в качестве контроллера домена (установлена роль **AD-Domain-Services**), DNS сервера (установлена роль **DNS**) и файлового сервера (установлены роли **FileAndStorage-Services**, **Storage-Services**). Большинство других компонентов используются для управления или мониторинга сервера.



Если вы точно не знаете имя роли, вы можете использовать подстановочные знаки. Например, чтобы проверить, какие веб-компоненты роли IIS установлены, выполните эту команду (синтаксис немного сокращён):

```
1 | Get-WindowsFeature -Name web-* | Where installed
```



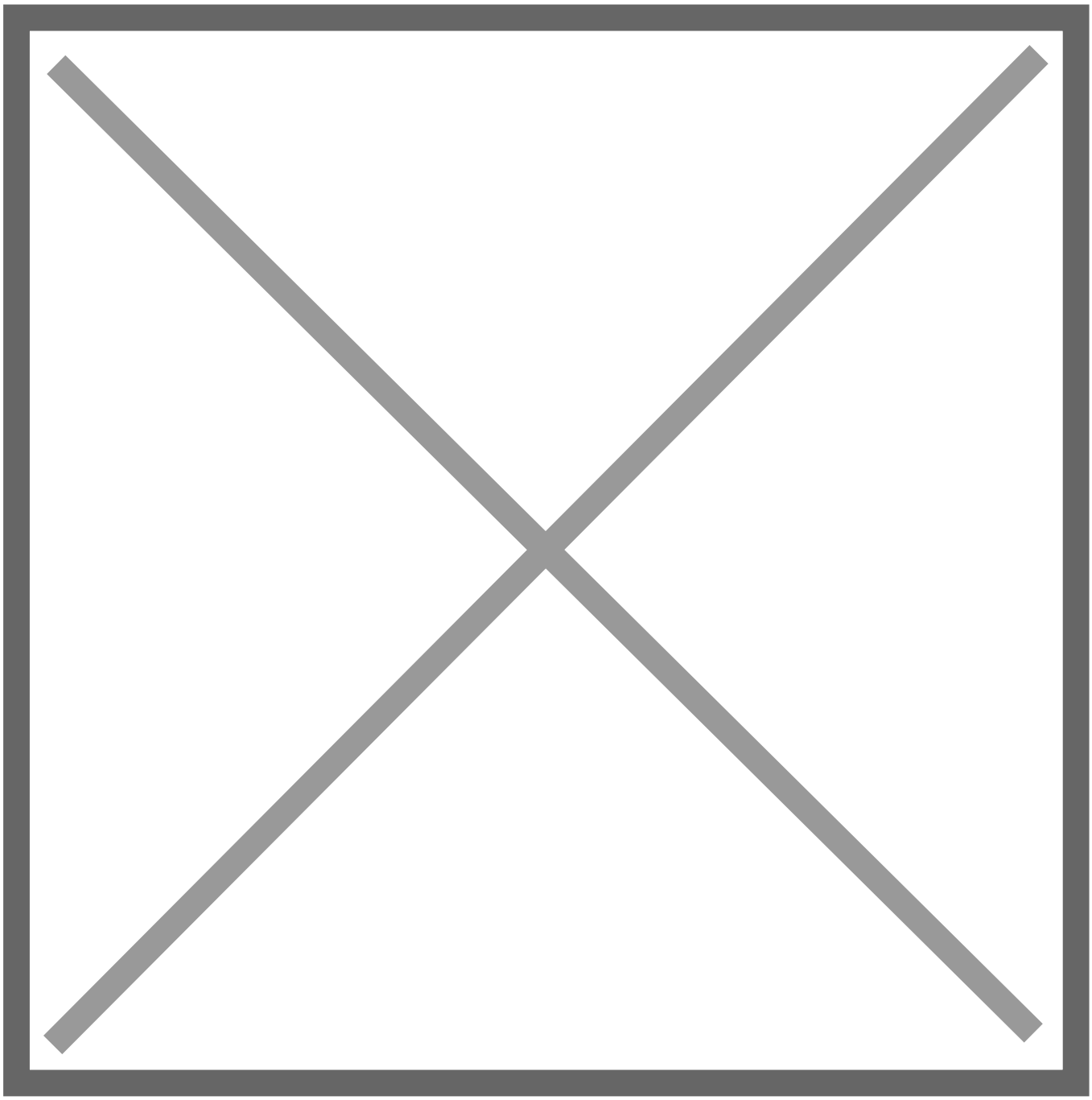
Вы можете получить список установленных компонентов на удалённом сервере Windows:

1	`Get-WindowsFeature -ComputerName ИМЯ_КОМПЬЮТЕРА | Where installed | ft Name,Installstate`

Вы можете использовать командлет **Get-WindowsFeature** для поиска серверов в вашем домене, на которых установлена определённая роль. Вы можете выполнять поиск на своих серверах в конкретном подразделении Active Directory с помощью командлета **Get-ADComputer** из модуля PowerShell ActiveDirectory или по предоставленному списку серверов **($ servers = ('server1', 'server2'))**.

Например, вы хотите найти все файловые серверы с ролью **FileAndStorage-Services** в указанном организационном подразделении AD, то используйте следующий скрипт:

```
Import-Module ActiveDirectory
$Servers=Get-ADComputer -Properties * -Filter {Operatingsystem -notlike "*2008 R2*" -and enabled -eq "true" -and Operatingsystem -like "*Windows Server*" } | select name
Foreach ($server in $Servers)
{
    Get-WindowsFeature -name FileAndStorage-Services -ComputerName $server.Name | Where installed | ft
$server.name, Name, Installstate
}
```



Пример поиска контроллера домена:

```
Import-Module ActiveDirectory
$Servers=Get-ADComputer -Properties * -Filter {Operatingsystem -notlike "*2008 R2*" -and enabled -eq "true" -and Operatingsystem -like "*Windows Server*" } | select name
Foreach ($server in $Servers)
{
    Get-WindowsFeature -name AD-Domain-Services -ComputerName $server.Name | Where installed | ft $server.name, Name, Installstate
}
```

В результате вы получите список серверов, на которых установлена конкретная роль.

Как с помощью PowerShell установить роли и компоненты Windows Server?

Для установки ролей и компонентов на Windows Server используется командлет Install-WindowsFeature.

Чтобы установить роль DNS-сервера и инструменты управления (включая модуль Powershell DNSServer) на текущий сервер, выполните следующую команду:

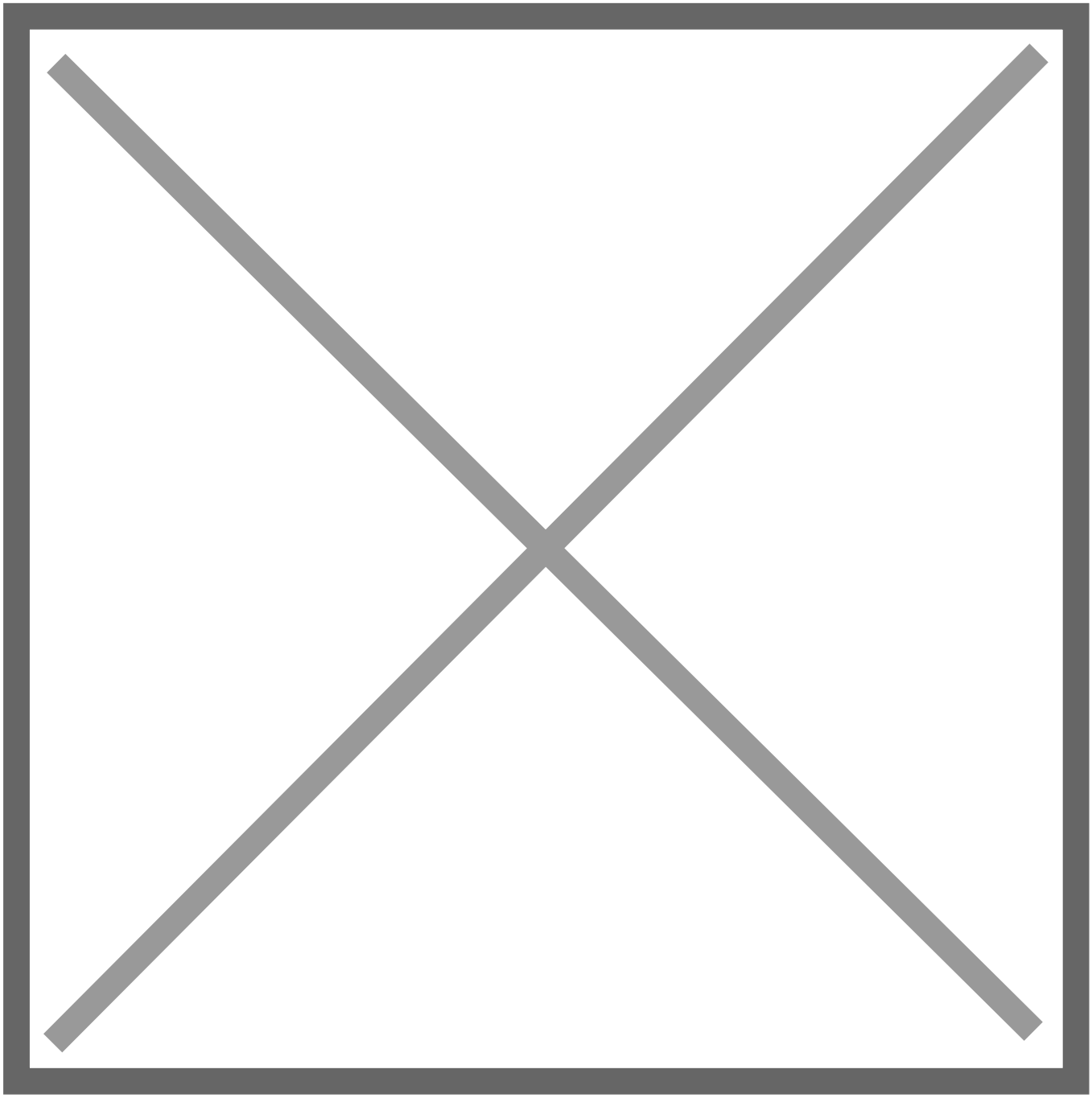
```
Install-WindowsFeature DNS -IncludeManagementTools
```

Установка роли «Active Directory Domain Services»:

```
Install-WindowsFeature -Name AD-Domain-Services -IncludeManagementTools
```

По умолчанию командлет устанавливает все зависимые роли и компоненты. Чтобы отобразить список зависимостей перед установкой, используйте опцию **-WhatIf**:

```
Install-WindowsFeature -Name UpdateServices -WhatIf
```



Чтобы установить роль узла сеанса удалённого рабочего стола, роль лицензирования RDS и инструменты удалённого управления RDS, используйте следующую команду:

```
1 Install-WindowsFeature -ComputerName ИМЯ_КОМПЬЮТЕРА RDS-RD-Server, RDS-Licensing -IncludeAllSubFeature -IncludeManagementTools -Restart
```

Если вы добавите параметр **-Restart**, ваш сервер при необходимости будет автоматически перезагружен.

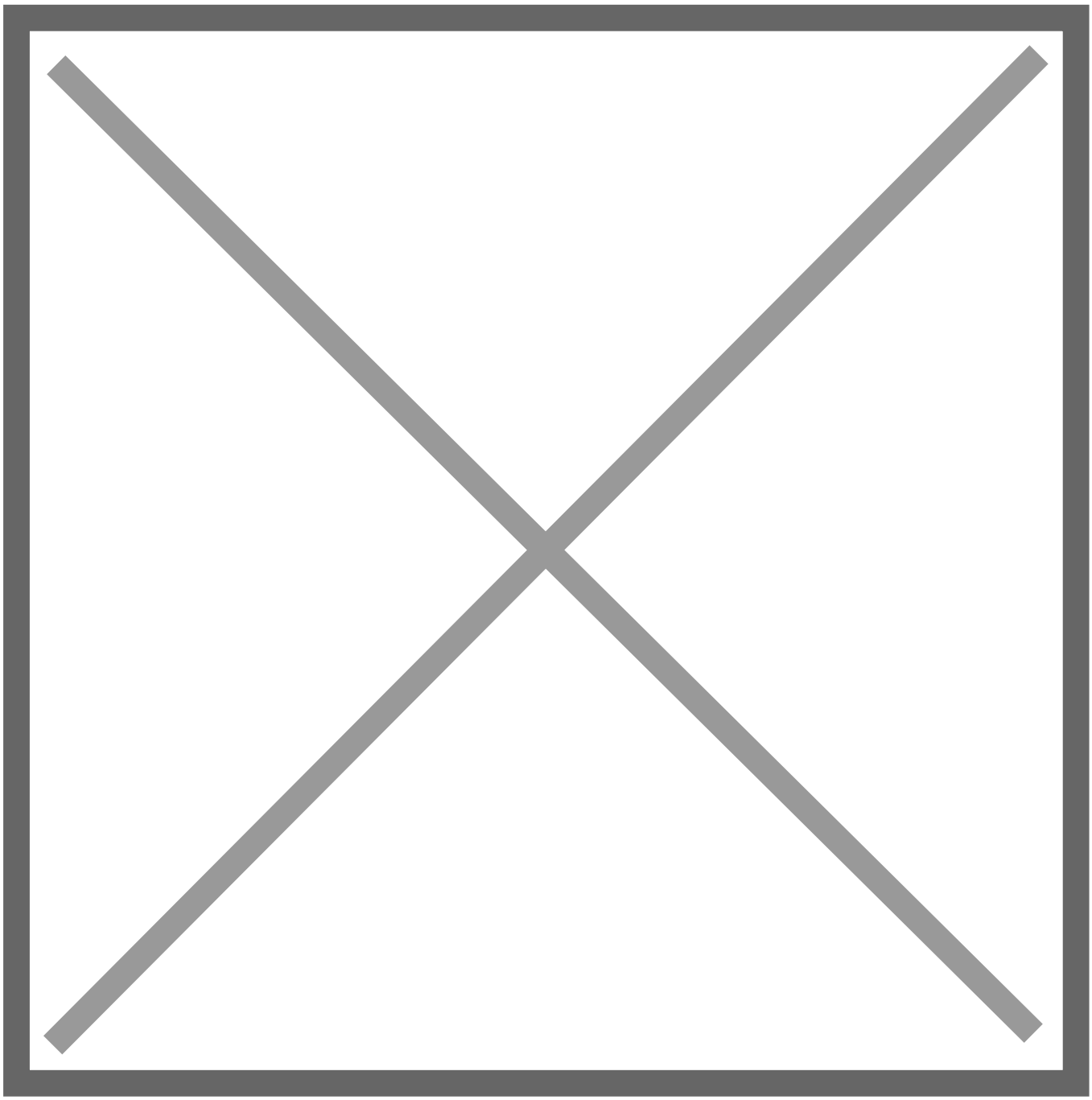
Вы также можете установить компонент с помощью следующей команды. Например, чтобы установить роль SMTP-сервера:

1	`Get-WindowsFeature -Name SMTP-Server | Install-WindowsFeature`

Как с помощью PowerShell развернуть роли на нескольких удалённых серверах Windows?

Есть ещё один интересный вариант при развёртывании типовых серверов. Вы можете установить необходимые функции на эталонном сервере Windows и экспортировать список установленных ролей в файл CSV:

1 2	`Get-WindowsFeature | where{$_.Installed -eq $True} | select name | Export-Csv C:\PS\InstalledRoles.csv -NoTypeInformation -Verbose` `Get-Content C:\PS\InstalledRoles.csv`



После этого вы сможете использовать этот CSV-файл для установки того же набора ролей на других типичных серверах:

1	`Import-Csv C:\PS\Roles.csv | foreach{ Install-WindowsFeature $_.name }`

Если роль или функция уже установлены, команда вернёт NoChangeNeeded и продолжит установку следующей роли.

Или, чтобы установить один и тот же набор ролей на нескольких удалённых серверах, вы можете использовать эту команду:

```
1
2
```

```
$servers = ( 'ny-rds1' , 'ny-rds2' ,'ny-rds3','ny-rds4')
foreach ($server in $servers) {Install-WindowsFeature RDS-RD-Server -ComputerName $server}
```

Как с помощью PowerShell удалить роль или компонент на Windows Server?

Чтобы удалить роли или функции Windows Server, используется командлет Remove-WindowsFeature.

Например, чтобы удалить роль сервера печати, выполните команду:

```
1
```

```
Remove-WindowsFeature Print-Server -Restart
```

Сертификат для Linux в центре сертификации Active Directory

Если в сети есть домен с контроллером и центр сертификации Microsoft, для внутреннего использования можно создавать собственные сертификаты, для которых браузер не будет выдавать ошибку. Если с запросом и установкой сертификата на Windows возникает меньше вопросов, то получение сертификата для Linux выполнить немного сложнее.

В этой статье мы создадим правильный сертификат с указанием альтернативного DNS-имени, без которого программы могут возвращать ошибку сертификата.

1. Формируем файл запроса (на компьютере с Linux)

Для начала создаем каталог, в котором планируем хранить сертификаты и перейдем в него, например:

mkdir -p /etc/ssl/adcs

cd /etc/ssl/adcs

Создаем закрытый ключ:

openssl genrsa -out private.key 2048

* *в данном примере создан **2048**-и битный ключ с именем **private.key***

Создаем файл с описанием запроса:

vi openssl.conf

[req]
default_bits = 2048

```
prompt = no
default_md = sha256
req_extensions = req_extensions
distinguished_name = dn

[dn]
C=RU
ST=SPb
L=SPb
O=Global Security
OU=IT Department
emailAddress=hostmaster@dmosk.ru
CN = *.dmosk.local

[req_extensions]
subjectAltName = @alter_name

[alter_name]
DNS.1 = *.dmosk.local
```

** где стоит обратить внимание на следующее:*

- ***sha256 (или SHA-2)** — алгоритм шифрования;*
- ****.dmosk.local** — имя узла, к которому мы будем обращаться. Это очень важный пункт — нужно, чтобы имя совпадало с именем, по которому будут выполняться обращения. В данном примере создается wildcart (домен и все его поддомены)*
- **subjectAltName** — альтернативные имена стали иметь значение с 2017 года, когда некоторые браузеры перестали принимать сертификаты без указания альтернативных DNS-имен.

Далее создаем ключ запроса сертификата:

```
openssl req -new -key private.key -out request.csr -config openssl.conf
```

** где **private.key** — закрытый ключ, который был сформирован на предыдущем шаге;*
***request.csr** — файл, который будет сформирован. В нем будет запрос на открытый ключ;*

После выполнения команды, в каталоге появится файл request.csr. Выведем его содержимое, чтобы посмотреть запрос:

```
cat request.csr
```

Должны увидеть что-то на подобие:

```
-----BEGIN CERTIFICATE REQUEST-----
MIIC1jCCAb4CAQAwgZAxCzAJBgNVBAYTAlJVMQwwCgYDVQQIDANTUGIxDDAKBgNV
BAcMA1NQYjEYMBYGA1UECgwPR2xvYmFsIFNlY3VyaXR5MRYwFAYDVQQLDA1JVCBE
```

ZXBhcnRtZW50MR4wHAYDVQQDDBVzaW1wbGVwbGFuLnNhdHMubG9jYWwxEzARBgNV
BAMMCnNpbXBsZXBsYW4wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDZ
ze/WOQgnlbfzlEsQUsmfUTnkYq0rCpzgjY360lFvqek5Y8NIFX/25PRbUy4N3D8r
c/7mXt2dXmcnn7zeRQOB2g0AY8Wmeg3R6C+JH7TwxtkMj7FO8R59URxFN84lu9Sj
26Aw+Ax7474XnAoUBMSmUXbV2mAP5Xm83sjvjE1OcHXN8SPbc+EchZuLVLsIGXHz
Emz7V4D/ecahfSc2hCRG2Pc7SeFIADYdjyoLtykz5WyiIoXkpEfSNQHlt2/A1kJ5
h9/GPXMVJVL02FgsI5HlGZyGnYWA+cP7sHoEDZNpLHHuEtfwx3bLxJPFnZDa0rPO
LhV/ux1e6b9ikrge0xp9AgMBAAGgADANBgkqhkiG9w0BAQsFAAOCAQEAPVD2aVH8
ZDz+6s8ecKr08eT3mA9cRCa7dZYFj4/vO/ZYrDH9QS45gd0sYG+RN+JMGaFaY+X7
faE4m0BysPIbhEYLRY5GJYxmOGm05gM2HfPrcnnWXZbPQu/n5pR4ptvPYL7bilGb
z6hXb8ZtXUXwz1F2OgTPOPu4+w8lI23pzHRHlCXcVSoZxe/A2XwusB5MrtyMEYtj
rB2kcOqQRkt9uIv5IobwnYaVGDk/7wl/zkb9K+RKZt4izKfFaSSyPn7wvpKSIaAf
S3SQjJ0tBckLbtwKrxdFB0B8bpyIKUtHmpX/zOmityC8PLXe6/vQ/DmM3B6QC4Ba
KdnRkOSPv8BGog==
-----END CERTIFICATE REQUEST-----

Чтобы проверить содержимое запроса, вводим:

openssl req -noout -text -in request.csr

Мы увидем что-то на подобие:

Certificate Request:
    Data:
        Version: 1 (0x0)
        Subject: C = RU, ST = SPb, L = SPb, O = Global Security, OU = IT Department, emailAddress = hostmaster@dmosk.ru, CN = *.dmosk.local
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (2048 bit)
                Modulus:
                    00:e0:dd:17:1c:e1:24:25:80:fe:a0:b3:39:95:bf:
                    9f:3b:a6:3f:62:c6:fa:40:19:0e:11:fe:9b:22:5b:
                    0c:67:57:08:b6:f4:36:4c:9c:aa:de:0b:ee:f2:11:
                    35:fc:18:36:ed:f5:e0:f9:82:98:bf:a5:1a:6f:a8:
                    13:20:2e:96:2a:68:13:db:e6:8e:0a:ce:fc:ee:c1:
                    fb:35:55:63:3f:bd:be:21:e7:f2:f3:fc:d3:b0:fd:
                    04:84:ce:84:29:2a:66:a9:ee:53:0e:f1:06:a2:b7:
                    0d:1c:8a:b9:e3:5d:ca:55:c3:30:77:48:d3:eb:d4:
                    27:11:8f:28:33:da:d4:4d:05:b6:da:f7:ea:84:3b:
                    8f:97:7d:26:f5:6f:09:39:8a:79:fc:6e:d2:3a:db:
                    4e:e8:67:2a:a7:22:00:a0:6c:c1:99:50:ea:3c:f7:
                    27:33:d3:5f:02:ed:7e:9e:66:17:fc:f3:af:f8:ef:
                    36:9e:da:3d:6c:c7:d3:b3:8b:4b:45:12:2f:84:34:
                    02:a5:0b:41:a1:6a:a3:91:6c:b0:87:d4:5b:cc:cc:

```
                f2:6c:17:4c:8b:46:40:65:5d:ca:8c:c0:f7:af:14:
                ad:58:1e:80:f5:67:16:1e:de:da:a5:7b:05:a6:f8:
                08:75:4b:8c:fe:53:e6:c8:50:00:30:ff:47:a5:49:
                47:c3
            Exponent: 65537 (0x10001)
    Attributes:
    Requested Extensions:
        X509v3 Subject Alternative Name:
            DNS:*.dmosk.local
Signature Algorithm: sha256WithRSAEncryption
    02:03:bc:8b:d3:4e:8f:2c:b2:a2:39:dd:9b:fb:33:a9:c1:49:
    95:b7:39:11:51:b2:40:dc:57:6c:3e:d5:66:6d:16:57:3f:15:
    03:cf:9b:39:95:1f:2b:13:09:86:f8:d5:69:b0:ac:cd:dc:64:
    9f:98:b7:f4:75:79:2a:16:d6:42:15:f4:1a:28:f3:3b:e3:f5:
    ec:34:98:f5:18:3e:72:95:ed:93:e5:e4:62:b3:3e:ce:71:09:
    9d:7f:ec:84:10:59:43:42:cd:0d:bd:4e:fd:c3:3e:44:ab:73:
    e0:ae:5d:67:c7:74:f0:30:0c:29:55:dc:16:4b:5f:a5:7f:be:
    8b:ec:64:3b:63:91:bb:48:d2:64:e5:8f:c0:09:db:7b:a3:10:
    8e:69:da:f7:88:00:a2:5a:60:dd:d9:3c:ef:b9:59:f4:b5:ab:
    00:d9:f9:31:6b:c1:c6:b1:ea:77:71:be:63:2c:e7:92:57:89:
    2a:5e:83:a9:e1:1c:56:c2:60:62:73:5d:bc:27:83:fb:bc:ca:
    9a:75:52:58:d6:02:d3:e8:37:b1:28:0e:89:62:97:9f:c4:f1:
    52:60:ed:95:59:2b:cd:69:29:9d:c2:30:1e:56:12:03:f4:17:
    3f:57:a2:b6:e7:b0:47:4e:0e:5a:b8:5e:0c:05:45:76:c5:49:
    f0:42:94:fa
```

2. Выпускаем сертификат

Копируем содержимое файла запроса (request.csr) и открываем веб-страницу центра сертификации — как правило, это имя сервера или IP-адрес + certsrv, например, http://192.168.0.15/certsrv/.

В открывшемся окне переходим по ссылкам **Запроса сертификата** - **расширенный запрос сертификата**.

В поле «Сохраненный запрос» вставляем скопированный запрос, в а поле «Шаблон сертификата» выбираем **Веб-сервер**:

Вводим данные для получения сертификата

Нажимаем **Выдать** и скачиваем сертификат по ссылке **Загрузить сертификат**:

Загрузка готового сертификата

3. Установка сертификата на Linux

Переносим полученный сертификат на компьютер с Linux.

После необходимо сконвертировать DER-сертификат (от AD CS) в PEM-сертификат (для Linux). Для этого вводим следующую команду:

openssl x509 -inform der -in certnew.cer -out public.pem

** где **certnew.cer** — файл, который мы получили от центра сертификации AD CS (как правило, у него такое имя); **public.pem** — имя PEM-сертификата.*

Сертификат готов к использованию. Мы сформировали:

1. private.key — закрытый ключ.
2. public.pem — открытый ключ.

Репликация Active Directory

Repadmin: How to Check Active Directory Replication

Example 2: Summarize the replication status and view the overall health

The first command you should use is replsummary. This command will quickly show you the overall replication health. This command will show you the percentage of replication attempts that have failed as well as the largest replication deltas.

```
repadmin /replsummary
```

Results displayed

```
:\WINDOWS\system32>repadmin /replsummary
Replication Summary Start Time: 2018-03-13 04:44:54

Beginning data collection for replication summary, this may take awhile:
  .....


Source DSA          largest delta    fails/total %%   error
 DC1                       52m:48s    0 /   5    0
 DC2                       52m:46s    0 /   5    0


Destination DSA     largest delta    fails/total %%   error
 DC1                       52m:46s    0 /   5    0
 DC2                       52m:48s    0 /   5    0
```

Example 3: Show replication partner and status

Next, use the following command to see the replication partner as well as the replication status. This helps you understand the role of each domain controller in the replication process.

In addition, this command displays the GUID of each object that was replicated and its result. This is helpful to identify what objects are failing to replicate.

```
repadmin /showrepl
```

Results displayed

```
C:\Users\rallen>repadmin /showrepl

Repadmin: running command /showrepl against full DC dc1.ad.activedirectorypro.com
Default-First-Site-Name\DC1
DSA Options: IS_GC
Site Options: (none)
DSA object GUID: a4d22a63-1918-492a-bcd6-7fe286941e72
DSA invocationID: a4d22a63-1918-492a-bcd6-7fe286941e72

==== INBOUND NEIGHBORS ======================================

DC=ad,DC=activedirectorypro,DC=com
    Default-First-Site-Name\DC2 via RPC
        DSA object GUID: 57a1cfbc-88bb-41da-a1a6-f14f5c9df408
        Last attempt @ 2018-03-13 03:52:08 was successful.

CN=Configuration,DC=ad,DC=activedirectorypro,DC=com
    Default-First-Site-Name\DC2 via RPC
        DSA object GUID: 57a1cfbc-88bb-41da-a1a6-f14f5c9df408
        Last attempt @ 2018-03-13 03:52:08 was successful.

CN=Schema,CN=Configuration,DC=ad,DC=activedirectorypro,DC=com
    Default-First-Site-Name\DC2 via RPC
        DSA object GUID: 57a1cfbc-88bb-41da-a1a6-f14f5c9df408
        Last attempt @ 2018-03-13 03:52:08 was successful.

DC=DomainDnsZones,DC=ad,DC=activedirectorypro,DC=com
    Default-First-Site-Name\DC2 via RPC
        DSA object GUID: 57a1cfbc-88bb-41da-a1a6-f14f5c9df408
        Last attempt @ 2018-03-13 03:52:08 was successful.

DC=ForestDnsZones,DC=ad,DC=activedirectorypro,DC=com
    Default-First-Site-Name\DC2 via RPC
        DSA object GUID: 57a1cfbc-88bb-41da-a1a6-f14f5c9df408
```

```
        Last attempt @ 2018-03-13 03:52:08 was successful.
```

Example 4: Show replication partner for a specific domain controller

If you want to see the replication status for a specific domain controller use this command.

replace <ServerName> with the name of your domain controller.

```
repadmin /showrepl <ServerName>
```

Results displayed

```
C:\WINDOWS\system32>repadmin /showrepl dc2
Default-First-Site-Name\DC2
DSA Options: IS_GC
Site Options: (none)
DSA object GUID: 57a1cfbc-88bb-41da-a1a6-f14f5c9df408
DSA invocationID: 2eb95693-bfa7-4f3f-b52c-139737aa883f

==== INBOUND NEIGHBORS ======================================

DC=ad,DC=activedirectorypro,DC=com
    Default-First-Site-Name\DC1 via RPC
        DSA object GUID: a4d22a63-1918-492a-bcd6-7fe286941e72
        Last attempt @ 2018-03-14 04:21:02 was successful.

CN=Configuration,DC=ad,DC=activedirectorypro,DC=com
    Default-First-Site-Name\DC1 via RPC
        DSA object GUID: a4d22a63-1918-492a-bcd6-7fe286941e72
        Last attempt @ 2018-03-14 03:52:07 was successful.

CN=Schema,CN=Configuration,DC=ad,DC=activedirectorypro,DC=com
    Default-First-Site-Name\DC1 via RPC
        DSA object GUID: a4d22a63-1918-492a-bcd6-7fe286941e72
        Last attempt @ 2018-03-14 03:52:07 was successful.

DC=DomainDnsZones,DC=ad,DC=activedirectorypro,DC=com
    Default-First-Site-Name\DC1 via RPC
        DSA object GUID: a4d22a63-1918-492a-bcd6-7fe286941e72
        Last attempt @ 2018-03-14 03:52:07 was successful.

DC=ForestDnsZones,DC=ad,DC=activedirectorypro,DC=com
    Default-First-Site-Name\DC1 via RPC
        DSA object GUID: a4d22a63-1918-492a-bcd6-7fe286941e72
        Last attempt @ 2018-03-14 03:52:07 was successful.
```

Example 5: Show only Replication Errors

The showrepl command can output a lot of information. If you want to see only the errors use this command. In this example, DC2 is down, you can see the results are all errors from DC2.

```
C:\WINDOWS\system32>repadmin /showrepl /errorsonly

Repadmin: running command /showrepl against full DC dc1.ad.activedirectorypro.com
Default-First-Site-Name\DC1
DSA Options: IS_GC
Site Options: (none)
DSA object GUID: a4d22a63-1918-492a-bcd6-7fe286941e72
DSA invocationID: a4d22a63-1918-492a-bcd6-7fe286941e72

==== INBOUND NEIGHBORS ======================================

DC=ad,DC=activedirectorypro,DC=com
    Default-First-Site-Name\DC2 via RPC
        DSA object GUID: 57a1cfbc-88bb-41da-a1a6-f14f5c9df408
        Last attempt @ 2018-03-15 04:19:38 failed, result 8524 (0x214c):
            The DSA operation is unable to proceed because of a DNS lookup failure.
        1 consecutive failure(s).
        Last success @ 2018-03-14 07:52:08.

CN=Configuration,DC=ad,DC=activedirectorypro,DC=com
    Default-First-Site-Name\DC2 via RPC
        DSA object GUID: 57a1cfbc-88bb-41da-a1a6-f14f5c9df408
        Last attempt @ 2018-03-15 04:19:38 failed, result 8524 (0x214c):
            The DSA operation is unable to proceed because of a DNS lookup failure.
        1 consecutive failure(s).
        Last success @ 2018-03-14 07:52:08.

CN=Schema,CN=Configuration,DC=ad,DC=activedirectorypro,DC=com
    Default-First-Site-Name\DC2 via RPC
        DSA object GUID: 57a1cfbc-88bb-41da-a1a6-f14f5c9df408
        Last attempt @ 2018-03-15 04:19:38 failed, result 8524 (0x214c):
            The DSA operation is unable to proceed because of a DNS lookup failure.
        1 consecutive failure(s).
        Last success @ 2018-03-14 07:52:08.

DC=DomainDnsZones,DC=ad,DC=activedirectorypro,DC=com
    Default-First-Site-Name\DC2 via RPC
        DSA object GUID: 57a1cfbc-88bb-41da-a1a6-f14f5c9df408
        Last attempt @ 2018-03-15 04:19:38 failed, result 8524 (0x214c):
            The DSA operation is unable to proceed because of a DNS lookup failure.
        1 consecutive failure(s).
        Last success @ 2018-03-14 07:52:08.

DC=ForestDnsZones,DC=ad,DC=activedirectorypro,DC=com
    Default-First-Site-Name\DC2 via RPC
```

```
        DSA object GUID: 57a1cfbc-88bb-41da-a1a6-f14f5c9df408
        Last attempt @ 2018-03-15 04:19:38 failed, result 8524 (0x214c):
            The DSA operation is unable to proceed because of a DNS lookup failure.
        1 consecutive failure(s).
        Last success @ 2018-03-14 07:52:08.

Source: Default-First-Site-Name\DC2
******* 1 CONSECUTIVE FAILURES since 2018-03-14 07:52:08
Last error: 8524 (0x214c):
            The DSA operation is unable to proceed because of a DNS lookup failure.
```

Example 6: Show replication Queue

It is normal to see items in the queue. If you have a small environment it will often be at zero because there are few replications that occur. If you notice items sitting in the queue and they never clear out, you have a problem.

Use this command to view the replication queue

```
Repadmin /Queue
```

Results displayed

```
C:\Users\rallen>repadmin /queue

Repadmin: running command /queue against full DC dc1.ad.activedirectorypro.com
Queue contains 0 items.
```

Example 7: How to Force Active Directory Replication

Use the following command if you want to force replication between domain controllers. You will want to run this on the DC that you wish to update. For example, if DC1 is out of sync I would run this on DC1.

This will do a pull replication, which means it will pull updates from DC2 to DC1.

```
repadmin /syncall dc1 /Aed
```

If you want to push replication you will use the /P switch. For example if you make changes on DC1 and want to replicate those to other DCs use this command.

```
repadmin /syncall dc1 /APed
```

Results displayed

```
C:\WINDOWS\system32>repadmin /syncall dc1 /Aed
Syncing all NC's held on dc1.
Syncing partition: DC=ForestDnsZones,DC=ad,DC=activedirectorypro,DC=com
CALLBACK MESSAGE: The following replication is in progress:
    From: 57a1cfbc-88bb-41da-a1a6-f14f5c9df408._msdcs.ad.activedirectorypro.com
    To  : a4d22a63-1918-492a-bcd6-7fe286941e72._msdcs.ad.activedirectorypro.com
CALLBACK MESSAGE: The following replication completed successfully:
    From: 57a1cfbc-88bb-41da-a1a6-f14f5c9df408._msdcs.ad.activedirectorypro.com
    To  : a4d22a63-1918-492a-bcd6-7fe286941e72._msdcs.ad.activedirectorypro.com
CALLBACK MESSAGE: SyncAll Finished.
SyncAll terminated with no errors.

Syncing partition: DC=DomainDnsZones,DC=ad,DC=activedirectorypro,DC=com
CALLBACK MESSAGE: The following replication is in progress:
    From: 57a1cfbc-88bb-41da-a1a6-f14f5c9df408._msdcs.ad.activedirectorypro.com
    To  : a4d22a63-1918-492a-bcd6-7fe286941e72._msdcs.ad.activedirectorypro.com
CALLBACK MESSAGE: The following replication completed successfully:
    From: 57a1cfbc-88bb-41da-a1a6-f14f5c9df408._msdcs.ad.activedirectorypro.com
    To  : a4d22a63-1918-492a-bcd6-7fe286941e72._msdcs.ad.activedirectorypro.com
CALLBACK MESSAGE: SyncAll Finished.
SyncAll terminated with no errors.

Syncing partition: CN=Schema,CN=Configuration,DC=ad,DC=activedirectorypro,DC=com
CALLBACK MESSAGE: The following replication is in progress:
    From: 57a1cfbc-88bb-41da-a1a6-f14f5c9df408._msdcs.ad.activedirectorypro.com
    To  : a4d22a63-1918-492a-bcd6-7fe286941e72._msdcs.ad.activedirectorypro.com
CALLBACK MESSAGE: The following replication completed successfully:
    From: 57a1cfbc-88bb-41da-a1a6-f14f5c9df408._msdcs.ad.activedirectorypro.com
    To  : a4d22a63-1918-492a-bcd6-7fe286941e72._msdcs.ad.activedirectorypro.com
CALLBACK MESSAGE: SyncAll Finished.
SyncAll terminated with no errors.

Syncing partition: CN=Configuration,DC=ad,DC=activedirectorypro,DC=com
CALLBACK MESSAGE: The following replication is in progress:
    From: 57a1cfbc-88bb-41da-a1a6-f14f5c9df408._msdcs.ad.activedirectorypro.com
    To  : a4d22a63-1918-492a-bcd6-7fe286941e72._msdcs.ad.activedirectorypro.com
CALLBACK MESSAGE: The following replication completed successfully:
    From: 57a1cfbc-88bb-41da-a1a6-f14f5c9df408._msdcs.ad.activedirectorypro.com
    To  : a4d22a63-1918-492a-bcd6-7fe286941e72._msdcs.ad.activedirectorypro.com
CALLBACK MESSAGE: SyncAll Finished.
SyncAll terminated with no errors.

Syncing partition: DC=ad,DC=activedirectorypro,DC=com
CALLBACK MESSAGE: The following replication is in progress:
    From: 57a1cfbc-88bb-41da-a1a6-f14f5c9df408._msdcs.ad.activedirectorypro.com
    To  : a4d22a63-1918-492a-bcd6-7fe286941e72._msdcs.ad.activedirectorypro.com
CALLBACK MESSAGE: The following replication completed successfully:
    From: 57a1cfbc-88bb-41da-a1a6-f14f5c9df408._msdcs.ad.activedirectorypro.com
    To  : a4d22a63-1918-492a-bcd6-7fe286941e72._msdcs.ad.activedirectorypro.com
```

```
CALLBACK MESSAGE: SyncAll Finished.
SyncAll terminated with no errors.
```

Example 8: Export results to text file

Sometimes these commands can display a lot of information. You can export any of the examples above to a text file, this makes it a little easier to review at a later time or save for documentation.

just add > c:\destination folder\filename.txt to the end of any of the commands

Here are a few examples

```
repadmin /replsummary > c:\it\replsummary.txt
```

```
repadmin /showrepl > c:\it\showrepl.txt
```

More examples

Find the last time your DC was backed up

```
Repadmin /showbackup *
```

Displays calls that have not yet been answered

```
repadmin /showoutcalls *
```

List the Topology information

```
repadmin /bridgeheads * /verbose
```

Inter Site Topology Generator Report

```
repadmin /istg * /verbose
```

Conclusion

As a system administrator, it is important that you know how to troubleshoot and verify replication is working correctly. The repadmin is a simple yet powerful tool that you should know how to use.

I hope you found this guide useful. If you have any questions leave a comment below. If you liked this article, check out: How to Use NSLookup to Check DNS Records.

Подпись файлов signtool.exe

> Требуется установить SDK - Windows Kits

C:\Program Files (x86)\Windows Kits\10\bin\10.0.19041.0\x64>signtool.exe sign /f C:\Users\baa\Desktop\PSO.pfx /p Qwerty123 /fd sha256 /tr http://timestamp.digicert.com /td sha256 C:\Users\baa\Desktop\pso.exe

/f C:\Users\baa\Desktop\PSO.pfx - Требуется указать путь до сертификата

/p Qwerty123 - Пароль серта

/fd sha256 - Указывает алгоритм хеширования файла

/tr http://timestamp.digicert.com - Указывает URL-адрес сервера меток времени

/td sha256 - Используется с `/tr` опцией `-` для запроса алгоритма, используемого сервером временных меток RFC 3161

C:\Users\baa\Desktop\pso.exe - Подписываемый файл