Windows Server Core: Remote management

kagarlickij.com/windows-server-core-remote-management

July 1, 2015

В предыдущей статье я рассказал про установку Windows Server Core, теперь о том, как управлять серверами развернутыми в core. Сервера, с которых будет выполнятся администрирование будем называть source, а сервера которые будем администрировать – target.

Target и source могут входить как в домен, так и в рабочую группу. Source может быть рабочим ПК администратора и работать под управлением Windows 7/8/8.1/10 с установленным пакетом RSAT соответствующей версии.

Рабочий ПК администратора не должен быть единственным местом откуда инфраструктурой можно управлять, его можно дополнить высокодоступным сервером размещенным в Microsoft Azure или Amazon, но их точкой отказа будет Интернет-канал.

Кроме RSAT, управлять серверами можно с помощью PowerShellWebAccess, но это скорее дополнительная возможность на случай недоступность RSAT. О настройке PSWA Вы можете почитать в моей статье “Настройка PowerShell Web Access“.

Перейдем непостредственно к настройке удаленного управления.

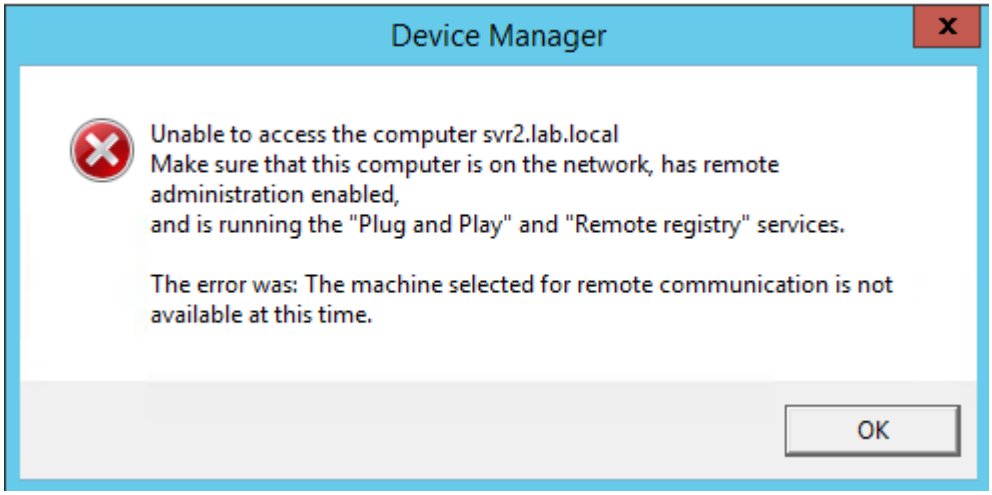
Чтобы посмотреть текущее значение удаленного управления на target можно выполнить:

```
Configure-SMRemoting.exe -get
```

Для удаленного управления, на целевом сервере должен быть настроен *WinRM*, его текущую конфигурацию можно запросить так:

```
winrm get winrm/config
```

Обратите внимание, Device Manager недоступен для удаленного управления в любых сценариях:

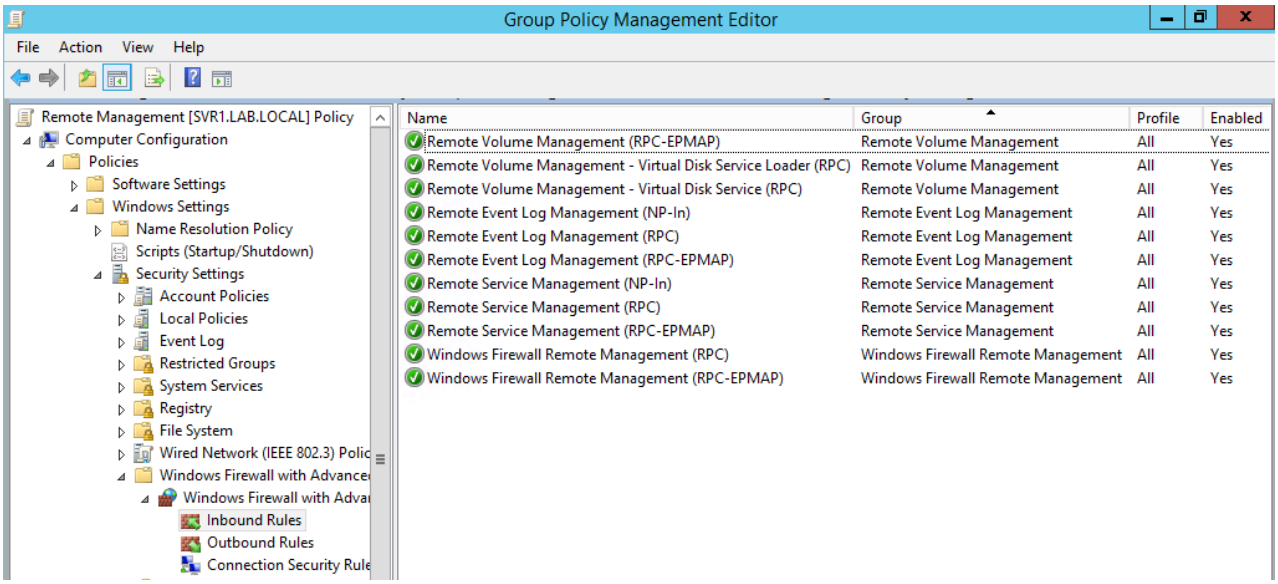


А все дело в том, что Microsoft “выпилила” удаленный доступ к PnP из соображений безопасности – http://support.microsoft.com/kb/2781106/en-us

Вместо этого, предлагается использовать PowerShell – http://blogs.technet.com/b/wincat/archive/2012/09/06/device-management-powershell-cmdlets-sample-an-introduction.aspx

Если Вам все-таки нужнен полноценный Device manager, вам придется установить хотя бы minimal GUI (о том, как это сделать написано выше).

Создадим и распространим на source и на target групповую политику, которой включим правила Firewall *Remote Event Log Management, Remote Service Management, Windows Firewall Remote Management, Remote Volume Management:*



В кажом правиле можно указать с каких сетей (лучше выделить админов в отдельную сеть, чем указывать список IP админских машин) разрешен этот трафик, на каких профилях и т.д. Хорошим вариантом будет использование IPSec.

Этот вопрос важен и требует индивидуального планирования чтобы, с одной стороны, минимизировать возможности атак, а с другой стороны, обеспечить возможность администрирования из нескольких мест, вт.ч. на случай

аварии.

Если вас интересует управление Windows Firewall с помощью PowerShell рекомендую эту статью.

Теперь рассмотрим сценарий когда source находится в домене, а target в рабочей группе. В начале нужно убедится что source и target корректно разрешают fqdn и netbios имена друг друга, если нет – нужно поправить это в DNS. Как и в большинстве случаев, предпочтительно использование fqdn имен.

После этого, на source нужно добавить имя target в TrustedHosts:

```
Set-Item WSMan:\localhost\Client\TrustedHosts -Value %target_fqdn% -Concatenate -
Force
```

После этого, можно будет использовать PowerShell remote sessions.

Вы можете посмотреть содержимое TrustedHosts:

```
Get-Item -Path WSMan:\localhost\Client\TrustedHosts | fl Name, Value
```

.. и очистить его содержимое при необходимости:

```
Clear-Item -Path WSMan:\localhost\Client\TrustedHosts
```

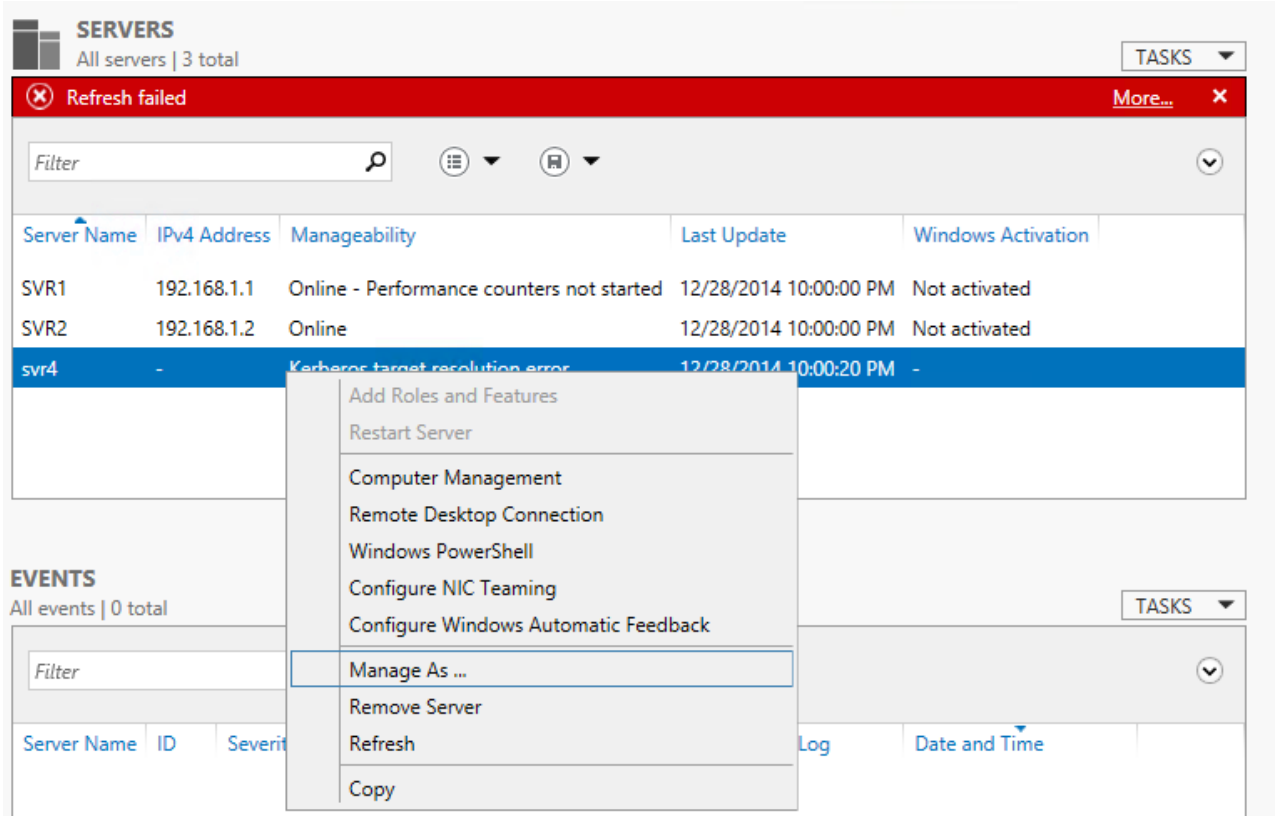
Теперь доступ к target есть по PowerShell, bus воспользуемся им чтобы включить на target таке правила firewall:

```
Set-NetFirewallRule -DisplayGroup "Windows Remote Management" -Enabled True -
RemoteAddress "192.168.1.0/24"

Set-NetFirewallRule -DisplayGroup "Remote Event Log Management" -Enabled True -
RemoteAddress "192.168.1.0/24"

Set-NetFirewallRule -DisplayGroup "Remote Service Management" -Enabled True -
RemoteAddress "192.168.1.0/24"

Set-NetFirewallRule -DisplayGroup "Windows Firewall Remote Management" -Enabled
True -RemoteAddress "192.168.1.0/24"

Set-NetFirewallRule -DisplayGroup "Remote Volume Management" -Enabled True -
RemoteAddress "192.168.1.0/24"

Set-NetFirewallRule -DisplayGroup "File and Printer Sharing" -Enabled True -
RemoteAddress "192.168.1.0/24"
```

Чтобы снять ограничения которые накладывает UAC на target нужно выполнить:

```
New-ItemProperty -Name LocalAccountTokenFilterPolicy -path
HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System -propertyType
DWord -value 1
```

Теперь можно добавить target в ServerManager на source и, а затем нужно выбрать опцию “*Manage As..*” и ввести учетные данные администратора target



Последний сценарий – когда source находится в рабочей группе, а target в домене – аналогичен предыдущему, и не требует дополнительных комментариев.

Если вам нужно управлять старыми версиями Windows Server, это сделать можно, 2012R2 и 2012 добавляются без проблем, а вот на 2008R2 и 2008 нужно будет поставить WMF 3.0 + Hotfix + выполнить:

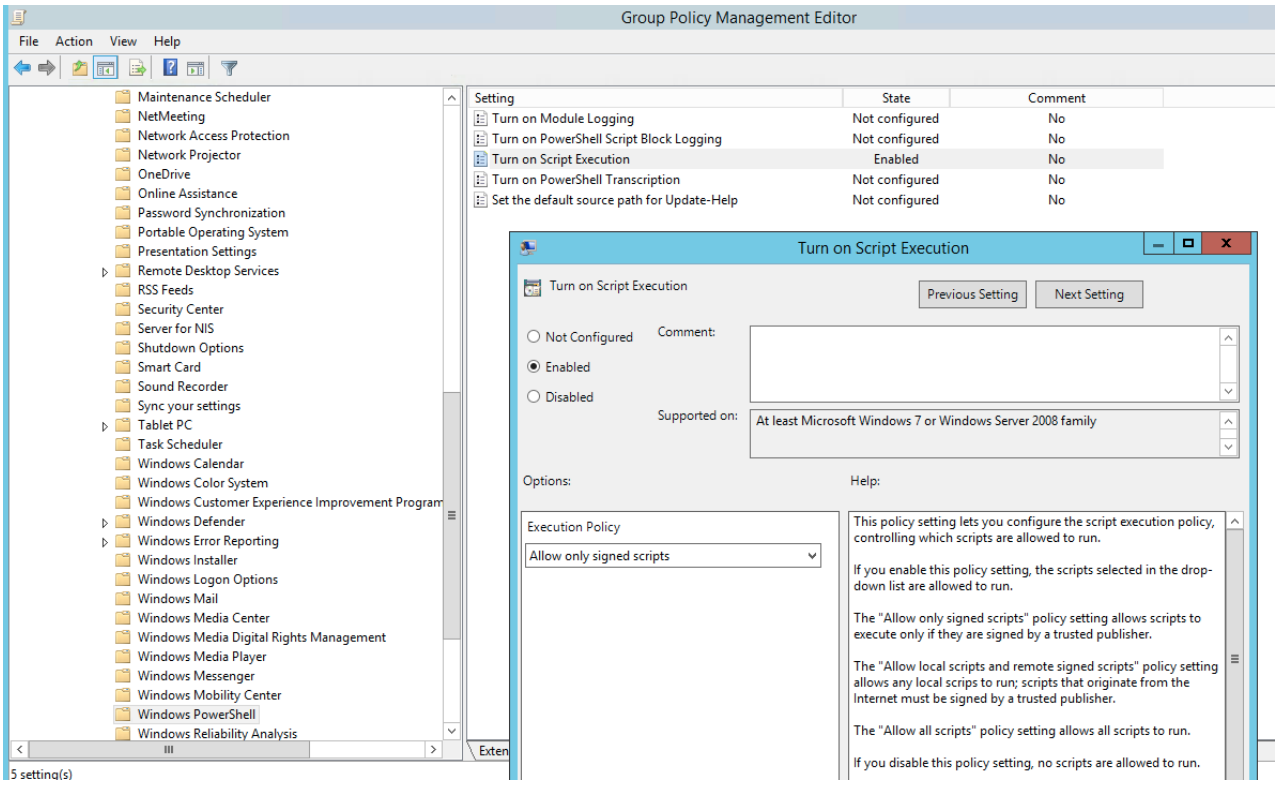
```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned
Enable-PSremoting -Force
```

После этого (само собой нужно включить правила firewall, о которых выше) серверами 2008 и 2008R2 можно будет ограниченно управлять, но нельзя, например, устанавливать роли и фичи.

Вообще, запускать скрипты без цифровой подписи в рабочей инфраструктуре не очень хорошо, поэтому есть смысл поставить политику AllSigned:

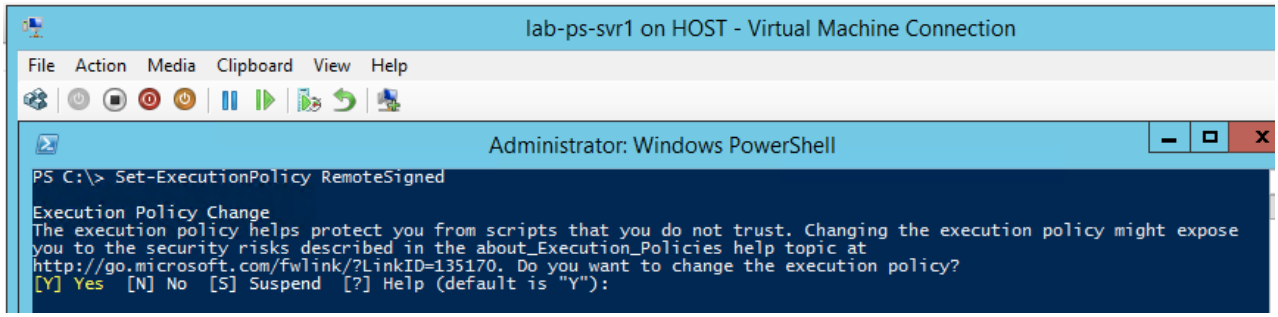
```
Set-ExecutionPolicy AllSigned
```

Если серверов больше чем несколько, есть смысл сделать это через групповую политику (Administrative Templates\Windows Components\Windows PowerShell):



Вот еще наглядный пример, почему большинство задач желательно выполнять через Remote Access:

```
[svr1]: PS C:\> Get-ExecutionPolicy
AllSigned
[svr1]: PS C:\> Set-ExecutionPolicy RemoteSigned
Security error.
    + CategoryInfo          : PermissionDenied: (:) [Set-ExecutionPolicy], SecurityException
    + FullyQualifiedErrorId : ExecutionPolicyOverride,Microsoft.PowerShell.Commands.SetExecutionPolicyCommand
```



Для подписи скриптов я буду использовать Comodo Code Signing certificate:

```
PS C:\> $MyCodeSigningCert = Get-ChildItem Cert:\CurrentUser\My -CodeSigningCert
PS C:\> $MyCodeSigningCert | Select-Object Issuer,FriendlyName,Version | Format-List


Issuer       : CN=COMODO RSA Code Signing CA, O=COMODO CA Limited, L=Salford, S=Greater Manchester, C=GB
FriendlyName : LEADCAPITAL MARKETS LIMITED
Version      : 3
```

Для подписывания используется командлет **Set-AuthenticodeSignature** :

```
PS C:\> Set-AuthenticodeSignature -FilePath C:\scripts\PSHelp_offline_update.ps1 -Certificate $MyCodeSigningCert


    Directory: C:\scripts

SignerCertificate                         Status                                 Path
-----------------                         ------                                 ----
646D93D44F4175909E70F211EE87339F5978285C  Valid                                  PSHelp_offline_update.ps1
```

Подписанный скрипт будет выглядеть следующим образом:

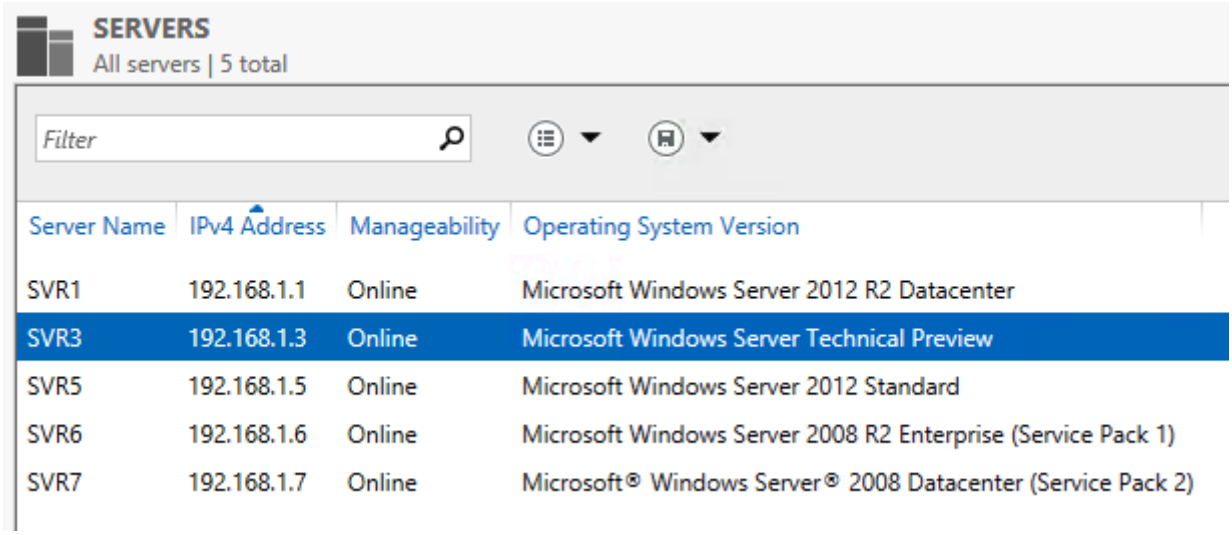
```
# Get result
if(-not $?) {
    $Result = "Failed"
    Write-EventLog -LogName Application -Source "PSHelp" -EntryType Warning -EventId 2 `
    -Message "PSHelp offline update $Result. TargetComputers: $TargetComputers"
} else {
    $Result = "Successful"
    Write-EventLog -LogName Application -Source "PSHelp" -EntryType Information -EventId 1 `
    -Message "PSHelp offline update $Result. TargetComputers: $TargetComputers"
}

# Send result to email
$From = "dmitriy@kagarlickij.com"
$To = "dmitriy@kagarlickij.com"
$Subject = "PSHelp offline update $Date $Result"

$Body = "PSHelp offline update $Date <br>"
$Body += "Result: <b>$Result</b> <br>"
$Body += "<br>"
$Body += "Updated computers: $TargetComputers <br>"

$SMTPServer = "smtp.office365.com"
$SMTPPort = "587"

$MailPass = Get-Content C:\scripts\MailboxSecurePass.txt | ConvertTo-SecureString
$MailCred = New-Object -TypeName System.Management.Automation.PSCredential `
-argumentlist "dmitriy@kagarlickij.com", $MailPass

Send-MailMessage -From $From -to $To -Subject $Subject `
-Body $Body -BodyAsHtml -SmtpServer $SMTPServer -port $SMTPPort -UseSsl `
-Credential $MailCred

# Variables cleanup
Remove-Variable -Name * -ErrorAction SilentlyContinue

# SIG # Begin signature block
# MIITuQYJKoZIhvcNAQcCoIITqjCCE6YCAQExCzAJBgUrDgMCGgUAMGkGCisGAQQB
# gjcCAQSgWzBZMDQGCisGAQQBgjcCAR4wJgIDAQAABBAfzDtgWUsITrck0sYpfvNR
# AgEAAgEAAgEAAgEAAgEAMCEwCQYFKw4DAhoFAAQUO06d9tFavXUW3J5giKjewZO6
# zUGgghDwMIIFdDCCBFygAwIBAgIQJ2buVutJ846r13Ci/ITeIjANBgkqhkiG9w0B
# AQwFADBvMQswCQYDVQQGEwJTRTEUMBIGA1UEChMLQWRkVHJ1c3QgQUIxJjAkBgNV
# BAsTHUFkZFRydXN0IEV4dGVybmFsIFRUUCBOZXR3b3JrMSIwIAYDVQQDExlBZGRU
# cnVzdCBFeHRlcm5hbCBDQSBSb290MB4XDTAwMDUzMDEwNDgzOFoXDTIwMDUzMDEw
# NDgzOFowgYUxCzAJBgNVBAYTAkdCMRswGQYDVQQIExJHcmVhdGVyIE1hbmNoZXN0
# ZXIxEDAOBgNVBAcTB1NhbGZvcmQxGjAYBgNVBAoTEUNPTU9ETyBDQSBMaW1pdGVk
# MSswKQYDVQQDEyJDT01PRE8gUlNBIENlcnRpZmljYXRpb24gQXV0aG9yaXR5MIIC
# IjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAkehUktIKVrGsDSTdxc9EZ3SZ
# KzejfSNwAHG8U9/E+ioSj0t/EFa9n3Byt2F/yUsPF6c947AEYe7/EZfH9IY+Cvo+
# XPmT5jR62RRr55yzhaCCenavcZDX7P0N+pxs+t+wgvQUfvm+xKYvT3+Zf7X8Z0Ny
# vQwA1onrayzT7Y+YHBSrfuXjbvzYqOSSJNpDa2K4Vf3qwbxstovzDo2a5JtsaZn4
# eEgwRdWt4Q08RWD8MpZRJ7xnw8outmvqRsfHIKCxH2XeSAi6pE6p8oNGN4Tr6MyB
# # SFNpIpIqm1y9IBsoilwie7SrmNnu4FGDwwlGTm0+mfqVF9p8M1dBPI1R7Qu2XK8s
```

При запуске необходимо будет принять решение по издателю, я обычно использую Run once – работая с большим количеством скриптов от коллег это становится необходимостью.

```
[svr1]: PS C:\> C:\scripts\PSHelp_offline_update.ps1

Do you want to run software from this untrusted publisher?
File C:\scripts\PSHelp_offline_update.ps1 is published by CN=LEADCAPITAL MARKETS LIMITED, OU=Dev, O=LEADCAPITAL MARKETS
 LIMITED, STREET="Treppides Tower, 9 Kafkasou Street,", STREET=5th Floor, L=Nicosia, S=-, PostalCode=2112, C=CY and is
not trusted on your system. Only run scripts from trusted publishers.
[V] Never run  [D] Do not run  [R] Run once  [A] Always run  [?] Help (default is "D"): _
```

Если вы используете для подписывания самозаверенный сертификат его нужно будет добавить на все сервера где планируется запуск подписанных им скриптов. Так что самозаверенные сертификаты, как всегда, лучше не использовать.

Добавление Windows Sever 2003 я не описываю т.к. во-первых в нем потолок PS 2.0, во-вторых его поддержка заканчивается в обозримом будущем, а в-третьих за годы его эксплуатации процессы управления наверняка налажены и менять их нецелесообразно.



Новый Server Manager сделал большой шаг перед на пути к выполнению массовых операций, но на практике PowerShell более функционален.

Команду на удаленном компьютере можно выполнить указав в Invoke-Command -ComputerName (по-умолчанию Invoke-Command добавляется ко всем командлетам выполняемым локально):

```
PS C:\> Invoke-Command -ComputerName svr1, svr2 {Get-Culture}

LCID             Name             DisplayName                    PSComputerName
----             ----             -----------                    --------------
1033             en-US            English (United States)        svr2
1033             en-US            English (United States)        svr1
```

Если команду нужно выполнить на нескольких сервера, откроем на них сессии и выполним операции на каждом параллельно:

```
PS C:\> $WebServersSessions = New-PSSession -ComputerName svr1, svr2, svr3
PS C:\> Invoke-Command -Session $WebServersSessions {Get-Service -Name W3SVC}

Status   Name               DisplayName                    PSComputerName
------   ----               -----------                    --------------
Running  W3SVC              World Wide Web Publishing Service svr3
Running  W3SVC              World Wide Web Publishing Service svr1
Running  W3SVC              World Wide Web Publishing Service svr2
```

Можете посмотреть разницу в скорости выполнения командлетов:

```
PS C:\> Measure-Command  { Invoke-Command { Get-ADGroup -Filter * } } | Select-Object Milliseconds | Format-Table -AutoS
ize

Milliseconds
------------
          17


PS C:\> Measure-Command  { Invoke-Command -ComputerName dc1 { Get-ADGroup -Filter * } } | Select-Object Milliseconds | F
ormat-Table -AutoSize

Milliseconds
------------
         812


PS C:\> Measure-Command  { Invoke-Command -Session $session { Get-ADGroup -Filter * } } | Select-Object Milliseconds | F
ormat-Table -AutoSize

Milliseconds
------------
          52


PS C:\> $session

 Id Name            ComputerName    State         ConfigurationName     Availability
 -- ----            ------------    -----         -----------------     ------------
  4 Session4        dc1             Opened        Microsoft.PowerShell     Available
```

Подробнее про управление:

http://technet.microsoft.com/en-us/library/hh831456.aspx

… старыми версиями –

http://blogs.technet.com/b/servermanager/archive/2012/09/10/managing-downlevel-windows-based-servers-from-server-manager-in-windows-server-2012.aspx

Надеюсь озвученная информация будет полезной, а если нужна будет помощь — используйте форму на главной странице моего сайта.