Powershell

Martin Čuchran

cuchran@muni.cz

Definícia

- „PowerShell is an automation platform and scripting language for Windows and Windows Server that allows you to simplify the management of your systems. Unlike other text-based shells, PowerShell harnesses the power of the .NET Framework, providing rich objects and a massive set of built-in functionality for taking control of your Windows environments.“

  Microsoft

- Súčasti:
  - Powershell
  - Powershell ISE
  - Powershell Desired State Configuration (DSC)

Obsah prednášky

- Background jobs
- Runspaces
- Debuging
- Remote management
- Novinky v Powershell 5.0

Motivácia

- Máte v správe 500 zamestnaneckých PC. Dostali ste hlásenie, že pri včerajších aktualizáciach SW sa korektne nenainštaloval požadovaný SW a naniektorých spravovaných PC chýbajú konfiguračné súbory. Ako zaistíte nápravu?

- Zabezpečujete správu identít v cloudovom prostredí Microsoft Office365 pre 500 zamestnancov. Všetkym užívateľom potrebujete nastaviť novú e-mailovu adresu pre ich účet. Aké možnosti máte?

Motivácia

- Máte v správe 500 zamestnaneckých PC. Dostali ste hlásenie, že pri včerajších aktualizáciach SW sa korektne nenainštaloval požadovaný SW a na niektorých spravovaných PC chýbajú konfiguračné súbory. Ako budete postupovať pri oprave?
  - 1. Zistím, na ktorých PC chýbajú požadované súbory:
    - Powershell remoting + Background jobs / runspaces
  - 2. Zaistím opravu
    - GPO + Powershell Remoting
- Zabezpečujete správu identít v cloudovom prostredí Microsoft Office365 pre 500 zamestnancov. Všetkym užívateľom potrebujete nastaviť novú e-mailovu adresu pre ich účet. Aké možnosti máte?

Motivácia

- Máte v správe 500 zamestnaneckých PC. Dostali ste hlásenie, že pri včerajších aktualizáciach SW sa korektne nenainštaloval požadovaný SW a na niektorých spravovaných PC chýbajú konfiguračné súbory. Ako budete postupovať pri oprave?
  - 1. Zistím, na ktorých PC chýbajú požadované súbory:
    - Powershell remoting + Background jobs / runspaces
  - 2. Zaistím opravu
    - GPO + Powershell Remoting
- Zabezpečujete správu identít v cloudovom prostredí Microsoft Office365 pre 500 zamestnancov. Všetkym užívateľom potrebujete nastaviť novú e-mailovu adresu pre ich účet. Aké možnosti máte a ako budete postupovať?
  - Použijem Web GUI a uklikam sa k smrti
  - Naskriptujem zmeny a použijem Powershell remoting pre komunikáciu s cloudom

Powershell Background jobs

- Uvedené vo verzii Powershell 2.0 (Windows server 2008r2<)
- Umožňuje asynchronný beh skriptov
- „Náhrada“ za vlákna
- Samostatná session pre každý job – vlastný proces
- Prístup len z rodičovského procesu

Powershell Background jobs

- Uvedené vo verzii Powershell 2.0 (Windows server 2008r2<)
- Umožňuje asynchronný beh skriptov
- „Náhrada“ za vlákna
- Samostatná session pre každý job – vlastný proces
- Prístup len z rodičovského procesu

Príklad:

```
Start-job –name Test –scriptBlock {Get-Process | select name}
Invoke-Command –computerName dc1 –scriptBlock {Get-Process} -asJob
```

Powershell Background jobs – cmdlety

- Spustenie jobu
  - **Start-Job -Name "Test Job" -ScriptBlock {Write-Output $args[0] > C:\output.txt}**
  - Sprostredkovanie argumentov pomocou „-argumentList“
- Získanie informácií o existujúcich Background job-och
  - **Get-Job -State ….**
- Prístup k výstupu z Background job-u
  - **Receive-Job**
  - Deštruktívne čítanie – $data = Receive-Job –Job ….
- Odstránenie Background job-u
  - **Remove-Job**
- Background job pomocou parametru „-asJob“
  - **Invoke-Command -ScriptBlock {….} -AsJob -ComputerName ....**

Powershell Background jobs – Príklad

- Zistite s využitím Powershell Background jobs prítomnosť súboru C:\file.txt na počítačoch zo zoznamu C:\computers.csv.

```
 #1
Get-Content C:\computers.csv | %{
    Invoke-Command -ComputerName $_
        -ScriptBlock {Write-output $args[0];Test-Path C:\file.txt}
        -ArgumentList $_
        -AsJob
}
while((Get-Job -State Running).count -gt 0){
    Start-Sleep -Seconds 1
}
Get-Job -State Completed | %{Receive-Job -Id $_.Id}
```

Powershell Background jobs – úloha

- Založte 5 skupín s názvom group<x>, kde x je z rozsahu <1,5> v Active Directory s využitím Powershellu background jobs. Výstup z operácie zapíšte do súboru C:\output.txt

- Zistite, či existujú súbory zo zoznamu files.csv na lokálnom PC s využitím Background-jobs.

Powershell Background jobs – úloha

- Založte 5 skupín s názvom group<x>, kde x je z rozsahu <1,5> v Active Directory s využitím Powershellu background jobs. Výstup z operácie zapíšte do súboru C:\output.txt

```
1..5 | %{
    Start-Job -ScriptBlock {New-ADGroup -Name $groupName -GroupCategory Security
           -GroupScope Global -Path "CN=Users,DC=pv176,DC=local"}
    -ArgumentList “group$($_)”
}
while((Get-Job -State Running).count -gt 0){
    Start-Sleep -Seconds 1
}
Get-Job -State Completed | %{Receive-Job -Id $_.Id} > C:\output.txt
```

- Zistite, či existujú súbory zo zoznamu files.csv na lokálnom PC s využitím Background-jobs.

Powershell Background jobs – úloha

- Založte 5 skupín s názvom group<x>, kde x je z rozsahu <1,5> v Active Directory s využitím Powershellu background jobs. Výstup z operácie zapíšte do súboru C:\output.txt

```
1..5 | %{
    Start-Job -ScriptBlock {New-ADGroup -Name $groupName -GroupCategory Security
           -GroupScope Global -Path "CN=Users,DC=pv176,DC=local"}
    -ArgumentList “group$($_)”
}
while((Get-Job -State Running).count -gt 0){
    Start-Sleep -Seconds 1
}
Get-Job -State Completed | %{Receive-Job -Id $_.Id} > C:\output.txt
```

- Zistite, či existujú súbory zo zoznamu files.csv na lokálnom PC s využitím Background-jobs.

```
Get-Content C:\files.csv | %{
    Start-Job -ArgumentList $_ -ScriptBlock {Write-Output $args[0]; Test-Path $args[0]}
}
while((Get-Job -State Running).count -gt 0){
    Start-Sleep -Seconds 1
}
Get-Job -State Completed | %{Receive-Job -Id $_.Id}
```

Literatúra

- https://msdn.microsoft.com/en-us/library/dd878288(v=vs.85).aspx
- https://msdn.microsoft.com/powershell/reference/5.1/Microsoft.PowerShell.Core/about/about_Jobs
- https://www.howtogeek.com/138856/geek-school-learn-how-to-use-jobs-in-powershell/

Powershell runspaces

- Alternatíva k Background jobs
- Výhody:
  - Rýchlejšia inicializácia
  - Menšie nároky na zdroje
  - Alternatíva k vláknam
- Nevýhody:
  - Nie je plnohodnotná Powershell session – nepodporuje všetky funkcie PS
  - Zložitosť – monitoring vlákna, ošetrenie chýb

Powershell runspaces – ukážka

```
$Runspace = [runspacefactory]::CreateRunspace()
$PowerShell = [powershell]::Create()
$PowerShell.runspace = $Runspace
$Runspace.Open()
[void]$PowerShell.AddScript({
    Get-Date
    Start-Sleep -Seconds 10
})
$AsyncObject = $PowerShell.BeginInvoke()
```

Literatúra

- https://blogs.technet.microsoft.com/heyscriptingguy/2015/11/26/beginning-use-of-powershell-runspaces-part-1/
- https://thesurlyadmin.com/2013/02/11/multithreading-powershell-scripts/

Powershell debugging

- Práca s výstupom
  - Write-Host, Write-Verbose, Write-Output, Write-Debug, Write-Error, Write-Warning
- Odchytávanie chýb
  - Try-Catch-Finally
- Logovanie
  - Start-transcript, Stop-transcript
  - Powershell logs

Powershell debugging - Práca s výstupom

- Práca s výstupom
  - Write-Host
  - Write-Verbose
  - Write-Output
  - Write-Debug
  - Write-Error
  - Write-Warning
  - Write-Information (PS 5.0)
- Aký je rozdiel medzi skupinami cmdletov “Write-host, Write-error, Write-warning” a “Write-Debug, Write-Verbose” ?

Powershell debugging - Práca s výstupom

- $DebugPreference
  - Continue
  - Stop
  - SilentlyContinue
  - Inquire
- $VerbosePreference
  - Continue
  - Stop
  - SilentlyContinue
  - Inquire
- $ErrorActionPreference - > Viac v ďalšej časti prednášky

Aký je rozdiel medzi jednotlivými hodnotami?

Môžeme zabezpečiť zobrazovanie debug a verbose výstupu aj iným spôsobom?

Powershell debugging - Práca s výstupom

- Príklad .\debugging.ps1
  - Čo vypíše na štandardný výstup zadaný script?

Powershell debugging - Práca s výstupom

- Príklad .\debugging.ps1
  - Čo vypíše na štandardný výstup zadaný script?
  - Ako zabezpečím aby sa na štandardný výstup vypísali všetky definované reťazce?

Literatúra

- https://msdn.microsoft.com/en-us/powershell/reference/5.1/microsoft.powershell.core/about/about_preference_variables
- https://msdn.microsoft.com/en-us/powershell/reference/5.1/microsoft.powershell.utility/write-host

Powershell debugging - Odchytávanie chýb

- Try-Catch-Finally

```
Try
{
    $AuthorizedUsers = Get-Content \\ FileServer\HRShare\UserList.txt -ErrorAction Stop
}
Catch [System.OutOfMemoryException]
{
    Restart-Computer localhost
}
Catch
{
    $ErrorMessage = $_.Exception.Message
    $FailedItem = $_.Exception.ItemName
    Send-MailMessage -From ExpensesBot@MyCompany.Com -To WinAdmin@MyCompany.Com -Subject "HR
File Read Failed!" -SmtpServer EXCH01.AD.MyCompany.Com -Body "We failed to read file
$FailedItem. The error message was $ErrorMessage"
    Break
}
Finally
{
    $Time=Get-Date
    "This script made a read attempt at $Time" | out-file c:\logs\ExpensesScript.log -append
}
```

Powershell debugging - Odchytávanie chýb

- Powershell pracuje s 2 typmi chýb
  - Terminating errors
  - Non-Terminating errors
- Try-Catch-Finally vyžaduje pre správnu funkčnosť aby cmdlety vyhadzovali Terminating errors

- Ako zabezpečíte, aby konštrukcia try-catch-finally vedela reagovať na akékoľvek chyby v rámci vášho skriptu?

Powershell debugging - Odchytávanie chýb

- Powershell pracuje s 2 typmi chýb
  - Terminating errors
  - Non-Terminating errors

- Ako zabezpečíte, aby konštrukcia try-catch-finally vedela reagovať na akékoľvek chyby v rámci vášho skriptu?
  - Konverzia Non-Terminating errors na Terminating errors
  - $ErroActionPreference
    - Continue
    - Stop
    - SilentlyContinue
    - Inquire
    - Suspend
  - -ErrorAction

Powershell debugging - Odchytávanie chýb

- Príklad .\Try-catch-finally.ps1
  - Čo vypíše na štandardný výstup zadaný script?

Powershell debugging - Odchytávanie chýb

- Príklad .\Try-catch-finally.ps1
  - Čo vypíše na štandardný výstup zadaný script?
  - Ako zabezpečím aby konštrukcia try-catch-finally fungovala správne?

Literatúra

- https://msdn.microsoft.com/en-us/powershell/reference/5.1/microsoft.powershell.core/about/about_preference_variables
- https://www.vexasoft.com/blogs/powershell/7255220-powershell-tutorial-try-catch-finally-and-error-handling-in-powershell

Powershell debugging - Logovanie

- S 5 kolegami spravujete 5000 PC. Ako zabezpečíte logovanie bezobslužných skriptov, ktoré sa spúšťajú na PC vo vašej správe?

Powershell debugging - Logovanie

- S 5 kolegami spravujete 5000 PC. Ako zabezpečíte logovanie bezobslužných skriptov, ktoré sa spúšťajú na PC vo vašej správe?
  - Využijem Write-* cmdlety, poriadne ošetrým odchytávanie výnimiek a donútim takéto riešenie používať všetkých kolegov

Powershell debugging - Logovanie

- S 5 kolegami spravujete 5000 PC. Ako zabezpečíte logovanie bezobslužných skriptov, ktoré sa spúšťajú na PC vo vašej správe?
  - Využijem Write-* cmdlety, poriadne ošetrým odchytávanie výnimiek a donútim takéto riešenie používať všetkých kolegov
  - Použijem Powershell Transcript

Powershell debugging - Logovanie

- S 5 kolegami spravujete 5000 PC. Ako zabezpečíte logovanie bezobslužných skriptov, ktoré sa spúšťajú na PC vo vašej správe?
  - Využijem Write-* cmdlety, poriadne ošetrým odchytávanie výnimiek a donútim takéto riešenie používať všetkých kolegov
  - Použijem Powershell Transcript
  - Zapnem logovanie pre Powershell pomocou GPO

Powershell debugging - Logovanie

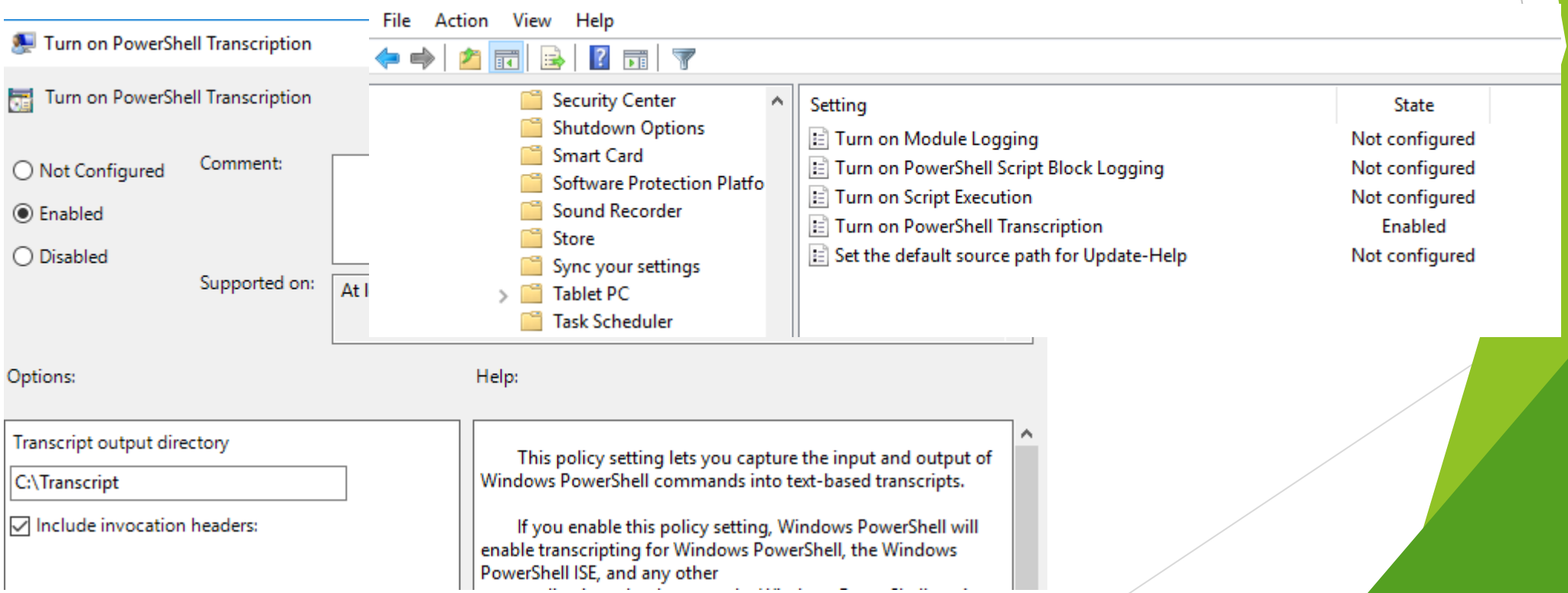
- Powershell Transcript
  - Zaznamenáva užívateľsky vstup a výstup z Powershell session do textového súboru. Záznam obsahuje aj aktuálne nastavenie Powershell session
- Powershell Transcript cmdlety
  - Start-Transcript
  - Stop-Transcript
- Powershell Transcript môže byť hromadne vynútený cez GPO
- Od verzie Powershell 5.0 dostupné aj v Powershell ISE

Powershell debugging - Odchytávanie výstupu

- Príklad: .\Transcript.ps1

Powershell debugging - Logovanie

- Nastavenie Powershell Transcript cez GPO
  - \Computer Configuration\ Administrative Templates\Windows Components\Windows PowerShell



Powershell debugging - Logovanie

- Úloha 1: Nastavte vytváranie transcript logov pre všetky PC v doméne pomocou GPO do adresáru C:\Transcript bez zaškrtnutej možnosti **„IncludeInvocationHeader“**
- Overte funkčnosť nastavenia pomocou spustenia Powershell príkazu na jednom z PC v doméne.
- Úloha 2: Nastavte vytváranie transcript logov pre všetky PC v doméne pomocou GPO do adresáru C:\TranscriptWithHeaders so zaškrtnutou možnosťou **„IncludeInvocationHeader“**
- Overte funkčnosť nastavenia pomocou spustenia Powershell príkazu na jednom z PC v doméne.

Aký je rozdiel vo vytvorených logoch?

Powershell debugging - Logovanie

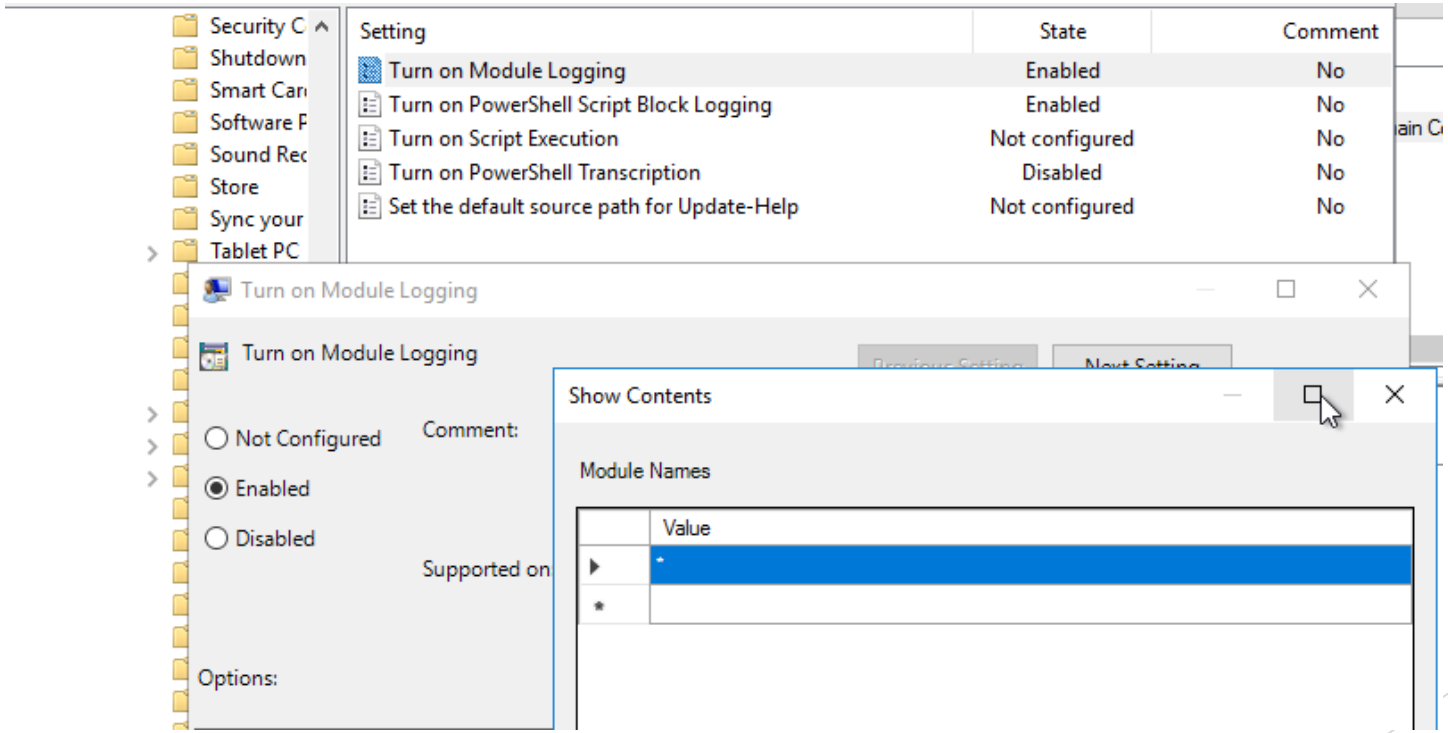
- Spracovanie Powershell príkazov je možné logovať aj pomocou systémoveho logu (Event viewer)
- Logovanie prebieha na 2 úrovniach:
  - Logovanie modulov (Event ID 4103)
  - Logovanie scriptov (Event ID 4104)
- Logy sú umiestnené v časti „Applications and Services Logs\Microsoft\Windows\PowerShell\Operational”
- Zapnutie pomocou GPO v \Computer Configuration\ Administrative Templates\Windows Components\Windows PowerShell
  - Turn on Module Logging
  - Turn on PowerShell Script Block Logging
- Zapnuté logovanie generuje netriviálne množstvo udalostí

Powershell debugging - Logovanie

- Úloha: Nastavte logovanie Powershell príkazov pre PC v doméne na úrovni modulov a scriptov pomocou GPO. Overte funkčnosť nastavenia.

Powershell debugging - Logovanie

- Úloha: Nastavte logovanie Powershell príkazov pre PC v doméne na úrovni modulov a scriptov pomocou GPO. Overte funkčnosť nastavenia.

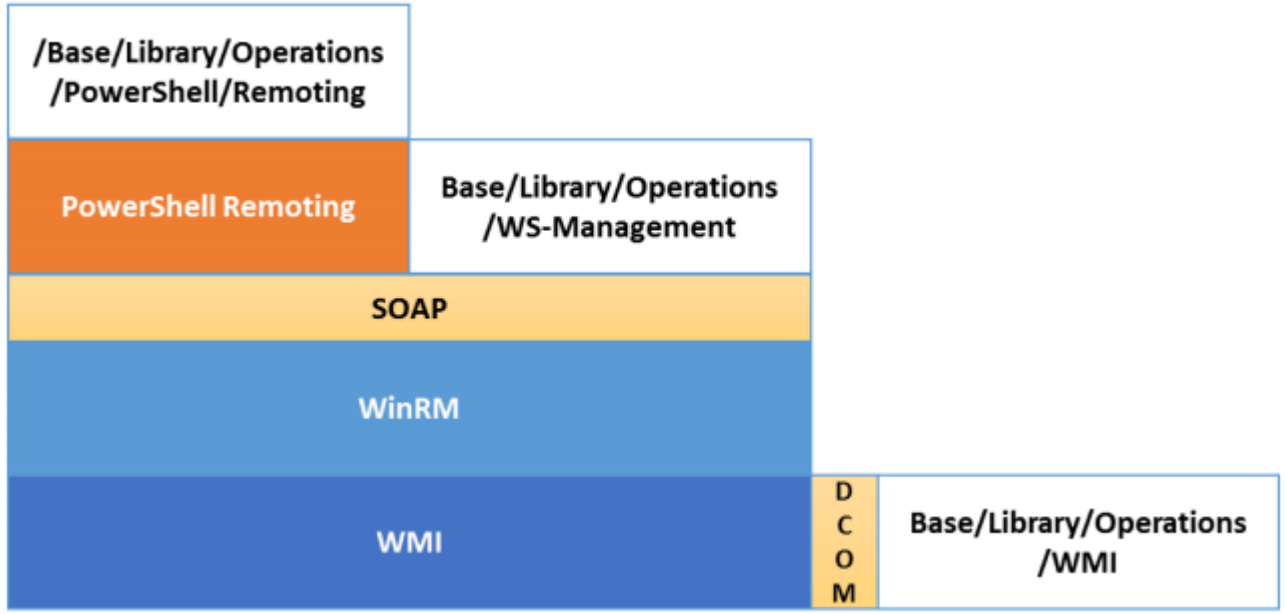


Literatúra

- https://www.fireeye.com/blog/threat-research/2016/02/greater_visibilityt.html
- https://msdn.microsoft.com/en-us/powershell/reference/5.1/microsoft.powershell.host/start-transcript

Powershell remote management

- V rámci prostredia Microsoft Powershell-u môžete využiť viacero technológií
  - RPC – Get-Service –ComputerName …
  - WMI – GetWMIObject –ComputerName …
  - WS-Management – Invoke-Command -ComputerName ….. (Powershell Remoting)



Powershell Remoting

- Využíva WS-Management
  - HTTP – TCP/5985
  - HTTPs – TCP/5986
- Autentizácia
  - Basic Auth
  - Digest
  - Kerberos
- Umožňuje vytvoriť
  - Persistentné spojenia
  - Interaktívne 1:1 spojenia
  - Spúšťať skripty na viacerých PC
- „Obdoba“ SSH

Powershell Remoting - Konfigurácia

- Powershell Remoting sprostredkováva služba WinRM
- WinRM pozostáva z klientskej a serverovej časti
- WinRM môžeme konfigurovať:
  - Lokálne pomocou Powershellu - **Enable-PSRemoting** a **winrm**
  - Pomocou GPO - Policies > Administrative Templates > Windows Components > Windows Remote Management (WinRM)
- Zobrazenie aktuálnej konfigurácie cez Powershell:

  winrm get winrm/config
- Základna konfigurácia

  winrm quickconfig a **winrm set winrm/config/….**

Powershell Remoting - Konfigurácia

- winrm quickconfig
- winrm get winrm/config
- winrm set winrm/config/client '@{AllowUnencrypted="true"}'

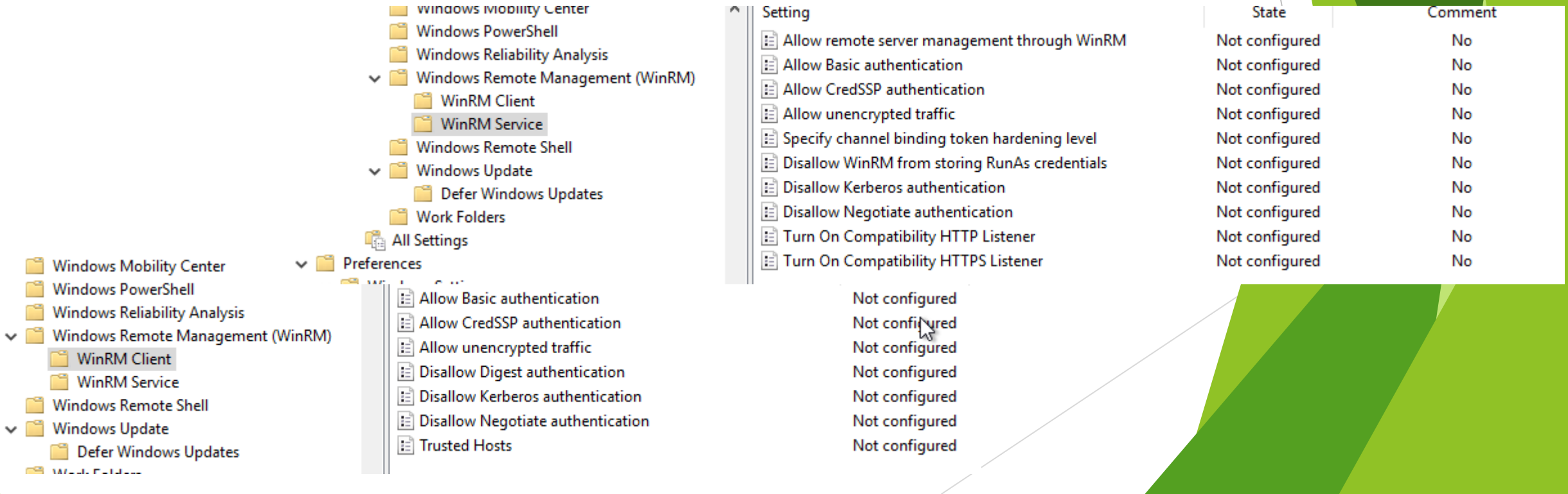
```
PS C:\Windows\system32> winrm quickconfig
WinRM is not set up to receive requests on this machine.
The following changes must be made:

Start the WinRM service.
Set the WinRM service type to delayed auto start.

Make these changes [y/n]? y

WinRM has been updated to receive requests.

WinRM service type changed successfully.
WinRM service started.
WSManFault
```

```
PS C:\Windows\system32> winrm get winrm/config
Config
    MaxEnvelopeSizekb = 500
    MaxTimeoutms = 60000
    MaxBatchItems = 32000
    MaxProviderRequests = 4294967295
    Client
        NetworkDelayms = 5000
        URLPrefix = wsman
        AllowUnencrypted = false
        Auth
            Basic = true
            Digest = true
            Kerberos = true
            Negotiate = true
            Certificate = true
            CredSSP = false
        DefaultPorts
            HTTP = 5985
            HTTPS = 5986
        TrustedHosts
    Service
        RootSDDL = O:NSG:BAD:P(A;;GA;;;BA)(A;;GR;;;IU)S:P(AU;FA;GA;;;WD)(AU;SA;GXGW;;;WD)
        MaxConcurrentOperations = 4294967295
        MaxConcurrentOperationsPerUser = 1500
        EnumerationTimeoutms = 240000
        MaxConnections = 300
        MaxPacketRetrievalTimeSeconds = 120
        AllowUnencrypted = false
        Auth
            Basic = false
            Kerberos = true
            Negotiate = true
            Certificate = false
            CredSSP = false
            CbtHardeningLevel = Relaxed
        DefaultPorts
            HTTP = 5985
            HTTPS = 5986
        IPv4Filter = *
        IPv6Filter = *
        EnableCompatibilityHttpListener = false
        EnableCompatibilityHttpsListener = false
        CertificateThumbprint
        AllowRemoteAccess = true
    Winrs
        AllowRemoteShellAccess = true
        IdleTimeout = 7200000
        MaxConcurrentUsers = 2147483647
        MaxShellRunTime = 2147483647
        MaxProcessesPerShell = 2147483647
        MaxMemoryPerShellMB = 2147483647
        MaxShellsPerUser = 2147483647
```

```
PS C:\Windows\system32> winrm set winrm/config/client '@{AllowUnencrypted="true"}'
Client
    NetworkDelayms = 5000
    URLPrefix = wsman
    AllowUnencrypted = true
    Auth
        Basic = true
        Digest = true
        Kerberos = true
        Negotiate = true
        Certificate = true
        CredSSP = false
    DefaultPorts
        HTTP = 5985
        HTTPS = 5986
    TrustedHosts
```

Powershell Remoting - Konfigurácia

- Pomocou GPO – Client + Service
  - Potrebné zabezpečit automatické spúšťanie služby WinRM pri štarte PC
  - Policies > Administrative Templates > Windows Components > Windows Remote Management (WinRM)



Powershell Remoting - Ukážka

Interaktívna Powershell session:

```
Enter-PSSession -ComputerName dc1
```

Persistentná Powershell session:

```
$session = New-PSSession -ComputerName dc1
```

Informácie o existujúcich Powershell sessions:

```
Get-PSSession
```

Ukončenie spojenia v prípade persistentnej Powershell session:

```
Remove-PSSession –Session $session
```

Spustenie scriptbloku v Powershell session:

```
Invoke-Command -Session $session -ScriptBlock {get-service -name dhcp}
```

Powershell Remoting - Úloha

- Zistite, či je na vašich doménovych radičoch povolený Powershell Remoting
- Zistite pomocou Powershell Remotingu z doménoveho radiča 1 aké služby bežia na doménovom radiči 2
- Zmeňte hodnotu „MaxConcurrentoperationsPerUser“ na 500 pre WinRM server na jednom z vašich radičov
- Vytvorte persistentné spojenie na PC v doméne pomocou Powershell Remotingu a doménového názvu PC. Zistite ako procesy bežia na vzdialenom PC
- Vytvorte persistentné spojenie na PC v doméne pomocou Powershell Remotingu a IP adresy PC. Na aký problém môžete naraziť?

Literatúra

- https://msdn.microsoft.com/en-us/library/aa384372(v=vs.85).aspx
- https://msdn.microsoft.com/powershell/reference/5.1/microsoft.powershell.core/Enable-PSRemoting
- https://blogs.technet.microsoft.com/josebda/2010/03/31/experimenting-with-powershell-v2-remoting/

Novinky v Powershell 5.0

- https://msdn.microsoft.com/en-us/powershell/scripting/whats-new/what-s-new-in-windows-powershell-50#BKMK_new50