TELEDIRIGIRAJMO POWERSHELLOM!

verzija 1.0

Ratko Žižek, MCSE-MCITP

1. Uvod

Važno je razumjeti da je PowerShell (od sada nadalje: **PShell**) strateški Microsoftov komandnolinijski admin alat i skriptni jezik, podjednako za Windows OS-ove koliko i za ostale ključne MS-ove produkte, spomenimo samo „najrazvikanije": Exchange, SharePoint, SQL, Forefront suita. Zbog MS-ove politike kontinuiranog reduciranja značaja i realnih mogućnosti GUI radnih površina u serverskim produktima s jedne strane te, s druge strane, neprekidnog bildanja PShella novim funkcionalnostima, Windows sistemci i developeri više ne smiju ignorirati ovu tehnologiju jer će stručno nazadovati.

Iako se **u članku fokusiramo isključivo na oblast udaljenog administriranja** Windows računala, na punih 5 stranica i tu smo temu samo načeli, toliko je PShell jak alat. Utoliko, sve što slijedi treba pojmiti kao smjernice za što brže ovladavanje temom – kako PShellom teledirigirano upravljati Windows računalima – a ne kao kompletno gradivo.

Do pojave PShella 2.0 Windows OS nije imao pravo rješenje za administriranje udaljenog računala u tekstualnom modu. Dakle, na tom je području Linux uvjerljivo prednjačio implementacijom SSH tehnologije. PShell 3.0, kojime su opremljeni Windows 2012 i Windows 8, značajno unaprijeđuje mogućnosti prethodne verzije, stoga nećemo pogriješiti tvrdnjom kako Windows konačno raspolaže kompletnim rješenjem za udaljeno administriranje Windows servera i radnih stanica u tekstualnom režimu rada.

Remotely se PShellom mogu administrirati sve windoze od dueta server 2003 / XP desktop na više. Uvijek aktualni, mi smo usmjereni na Windows Server 2012 / Windows 8, time ujedno na PShell 3.

2. PRIPREMA RAČUNALA ZA UDALJENO ADMINISTRIRANJE

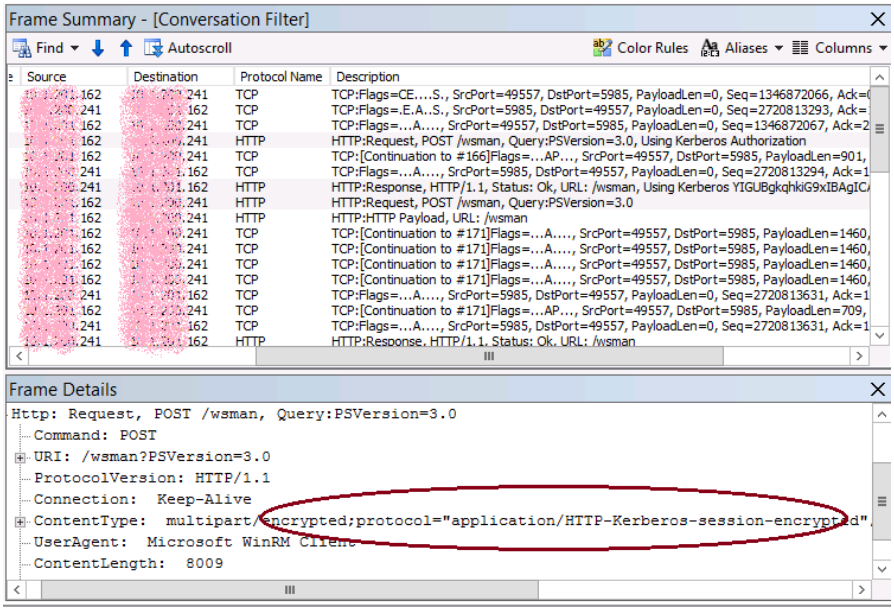
Slijedi nekoliko bitnih informacija o WinRM servisu, o kojemu PShell ovisi, potom i o samom PShellu, ali isključivo u kontekstu *remote managementa.*

2. 1. WinRM servis

- WinRM servis mora biti startan; mudro je tada postaviti mu Startup Type na Automatic.
- Ako su računala u domeni, s WinRM-om nemamo puno posla; kao transportni protokol rabi se HTTP, autentikacijski protokol je Kerberos; promet je enkriptiran, uključujući i etapu autentikacije (vidi sliku 1).

Slika 1:

Vizualna potvrda gornje informacije.



- Kad su admin i ciljno računalo u domeni, a pristupamo računalu-cilju preko nesigurne mreže, mudro je prije spajanja PShellom podići VPN tunel ili IPSec enkripciju; moguće je rješenje prijelaz na HTTPS pa tada nemamo potrebu štititi PShell sesije (o tome vidi iduću točku).
- Ukoliko su računala izvan domenskog prostora, WinRM servis moramo rekonfigurirati, u protivnom se PShell sesije neće moći uspostaviti; također, neophodno je prijeći na HTTPS, što povlači i primjenu certifikata za računala. Procedura je objašnjena pod linkom http://blogs.technet.com/b/meamcs/archive/2012/02/25/how-to-force-winrm-to-listen-interfaces-over-https.aspx.
- Lokalni firewall mora omogućiti pristup ka portovima na kojima sluša WinRM *listener*, to su TCP 5985 (za HTTP konekciju) ili 5986 (HTTPS).
- Stanje servisa može se provjeriti iz PShell konzole naredbom **get-service winrm** (lokalno) odn. sa **get-service winrm –computername ime-cilja** za udaljeno računalo.
- Detaljne info o aktualnoj konfiguraciji servisa dat će nam **winrm get winrm/config** (zadati iz PShella).

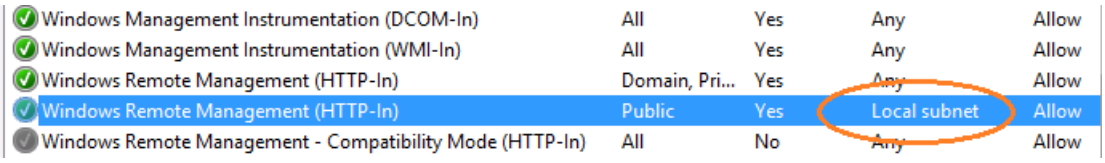
2.2. PowerShell

- Admin stanica i računalo-cilj moraju imati .NET Framework 4.0 te WinRM 3.0 i PowerShell 3.0 kao minimalne verzije tog SW-a; srećom, Windows Server 2102 i Windows 8 u startu su opremljeni baš tim verzijama.
- Lakše je konfigurirati računalo i rabiti PShell *remote* značajku kad su računala u domeni. Rečeno **ne** implicira da moraju biti na istoj mreži.
- Windows Server 2012 dolazi podešen za *remote management* PShellom. To nije slučaj s Windows 8 pa, ukoliko planiramo i nju teledirigirano administrirati, moramo ju pripremiti naredbom **enable-psremoting**.
- Povezano s gornjim: da bi ciljno računalo prihvaćalo PShell sesije inicirane s admin stanice, na ciljnom računalu otvorimo PShell konzolu i naredimo **enable-**

psremoting (dodamo li opciju **–force**, izbjeći ćemo par pitanja); ovime smo ujedno podesili i lokalni firewall za propuštanje konekcija s privatnih mreža.

Enable-PSRemoting spada u naredbe koje se izvršavaju samo ako je PShell konzola pokrenuta u Administrator security kontekstu. Na Core ediciji se konzola defaultno pokreće u tom kontekstu ali na GUI ediciji (Windows Server 2012 i Windows 8) treba iskoristiti naredbu Run as Administrator. Što se tiče lokalnog accounta Administrator, on može, kako se već godinama preporuča, biti disabliran.

- Ako baš želimo otvarati sesije PShellom i s *untrusted* (public) mreža, moramo na računalu-cilju još zadati **Set-NetFirewallRule –Name "WINRM-HTTP-In-TCP-PUBLIC" –RemoteAddress Any**



Slika 2 jasno pokazuje zašto je potreban dodatni zahvat u lokalni firewall za otvaranje pristupa s Public mreže.

- Podešenost računala za prihvat konekcija tipa *remote* provjeravamo tako da na njemu, dakako, u PShell konzoli, zadamo **new-pssession**.

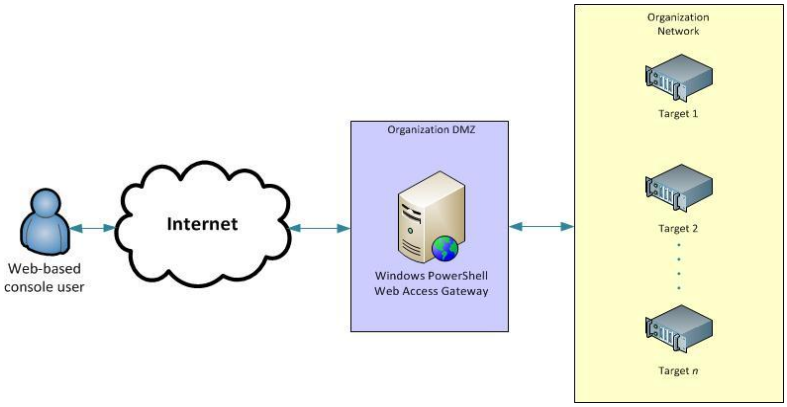
```
PS C:\> new-pssession

 Id Name            ComputerName    State         ConfigurationName     Availability
 -- ----            ------------    -----         -----------------     ------------
  1 Session1        localhost       Opened        Microsoft.PowerShell     Available
```

Slika 3: Ovo računalo je podešeno za udaljeno administriranje PShellom.

- Tijekom spajanja na ciljno računalo možemo rabiti domenski ili lokalni account koji ima odgovarajuće ovlasti na ciljnom računalu, **ali** najveći radni komfort imat ćemo samo ako i na polazištu i na cilju rabimo isti admin **domenski** account.
- **PowerShell Web Access** je zanimljiva implementacija PShella jer omogućuje udaljenu obradu Windows računala PShell naredbama i skriptama s raznih uređaja (npr. kiosk-računala, tableta...) bez potrebe za lokalnim WinRM i PShell instalacijama. Nužni su preduvjeti PowerShell Web Access Gateway server (na kojem nas, dakako, čeka PShell radno okruženje) te s PWA tehnologoijom kompatibilan browser na uređaju - PWA klijentu. Koncept je kvalitetno obrađen na http://technet.microsoft.com/en-us/library/hh831611.aspx. Uočite da se, primjenom ove tehnologije, za daljinsku kontrolu Windows računala PShellom može rabiti i, recimo, Linuxom pogonjeni tablet.

*Ovu sliku neću numerirati jer sam ju, psssst... **ukrao** s gore spomenutog linka; odlično ilustrira PWA koncept.*



3. TIPOVI POWERSHELL REMOTE SESIJA

3.1. Postoje naredbe koje dopuštaju navođenje ciljnog računala odn. ciljnih računala opcijom **–computername**. Popis takvih naredbi dobijemo sa **get-command –parametername computername**. Na nižoj slici rabimo jednu od naredbi iz tog skupa da bismo provjerili zakrpe na dva računala.

```
PS C:\> Get-HotFix -ComputerName srvrtm2, win8pro1

Source        Description      HotFixID      InstalledBy          InstalledOn
------        -----------      --------      -----------          -----------
SRVRTM2       Update           KB2751352     NT AUTHORITY\SYSTEM  9.12.2012. 0:00:00
WIN8PRO1      Update           KB2751352     NT AUTHORITY\SYSTEM
WIN8PRO1      Update           KB976002      NT AUTHORITY\SYSTEM


PS C:\>
```

*Slika 4: Uočite da u naredbi opremljenoj parametrom **–computername** možemo navoditi više ciljnih računala.*

3.2. Naredbe koje nemaju parametar **-computername** ipak možemo iskoristiti u remote sesiji posredstvom **invoke-command** naredbe. Slijedi par primjera.

PS C:\> invoke-command –computername srvrtm2 –scriptblock {get-windowsedition –online} = saznat ćemo koja je edicija windoza na udaljenom Srvrtm2 serveru

PS C:\pscripts> invoke-command -computername corertm1 -filepath .\netdiag.ps1 = skripta netdiag.ps1, koja se nalazi na admin stanici, u direktoriju c:\pscripts (i admin je u tom direktoriju, vidi prompt), izvršit će se na udaljenom računalu Corertm1 i prikazati rezultat na ekranu

3.3. Zasebna su skupina ***-pssession** naredbe, eno ih na slici 5. Ovim naredbama upravljamo tzv. interaktivnim i perzistentnim sesijama.

*Slika 5 pokazuje dva bitna skupa naredbi za remote sesije, upravljački i konfiguracijski. Informativniji ispis dobit ćemo ako zamijenimo **get-command** sa **get-help**.*

```
PS C:\> Get-Command *-pssession

CommandType     Name
-----------     ----
Cmdlet          Connect-PSSession
Cmdlet          Disconnect-PSSession
Cmdlet          Enter-PSSession
Cmdlet          Exit-PSSession
Cmdlet          Export-PSSession
Cmdlet          Get-PSSession
Cmdlet          Import-PSSession
Cmdlet          New-PSSession
Cmdlet          Receive-PSSession
Cmdlet          Remove-PSSession


PS C:\> get-command *-PSSessionConfiguration

CommandType     Name
-----------     ----
Cmdlet          Disable-PSSessionConfiguration
Cmdlet          Enable-PSSessionConfiguration
Cmdlet          Get-PSSessionConfiguration
Cmdlet          Register-PSSessionConfiguration
Cmdlet          Set-PSSessionConfiguration
Cmdlet          Unregister-PSSessionConfiguration


PS C:\> _
```

INTERAKTIVNA se sesija uspostavlja sa (osnovni oblik):
enter-pssession -computername ime-cilja

Riječ je o „telnet-alike" tipu komunikacije, naime, iz prompta se vidi da smo na ciljnom računalu, sve naredbe izvršavaju se na tom računalu, itd. Za zatvaranje takve sesije je **exit-pssession** ili, kraće, **exit**. Za nas sistemce ovo je uobičajeni, rekli bismo svakodnevni način rada.

```
PS C:\> Enter-PSSession corertm2
[corertm2]: PS C:\Users\ratko\Documents> cd \
[corertm2]: PS C:\> get-process -module powershell > c:\ps-proces.txt
[corertm2]: PS C:\> new-psdrive -name R -PSProvider FileSystem -root \\corertm1\systools -credential [redacted]\ratko

Name           Used (GB)     Free (GB) Provider      Root                                               CurrentLocation
----           ---------     --------- --------      ----                                               ---------------
R                                      FileSystem    \\corertm1\systools


[corertm2]: PS C:\> copy-item .\ps-proces.txt R:
[corertm2]: PS C:\> exit
PS C:\>
```

Slika 6 prikazuje tipičnu remote sesiju tj. s admin stanice virtualno smo prešli na jedan Core server i od tog trenutka radimo na njemu: u C:\ spremili smo datoteku sa podacima o aktualnom PowerShell procesu, spojili se na mrežni direktorij drugog Core servera i u njega kopirali datoteku te, naposljetku, prekinuli sesiju.

PERZISTENTNA sesija na udaljeno računalo otvara se sa (osnovni oblik):
new-pssession –computername ime-cilja

Ovaj tip sesije osnovica je uporabe raznih naprednih funkcionalnosti PShella, od uspostavljanja interaktivne sesije kroz postojeću permanentnu (naredba **enter-pssession** ima parametar **–session** baš za tu svrhu), preko izlaska iz sesije na jednom računalu da bismo satima kasnije ušli u tu sesiju s drugog računala (moguće jer o permanentnoj sesiji brigu vodi PShell Session Manager na ciljnom računalu), sve do preusmjeravanja funkcija, varijabli, podataka... iz jednog zadatka (naredbe, skripte) u drugi, itd. Zbog posljednje spomenute značajke mudro je kontrolirati broj permanentnih sesija na jedno računalo-cilj jer svaka takva sesija od sustava traži i ponešto od ključnih resursa za sebe.

```
PS C:\> $rale = New-PSSession -ComputerName corertm2, srvrtm2, win8pro1
PS C:\> Invoke-Command -Session $rale {get-smbshare}
```

Slika 7 pokazuje jedan od jednostavnijih, time i razumljivijih, načina uporabe perzistentne sesije: sesiju uspostavljamo sa tri računala ali značajke te sesije ujedno spremamo u varijablu $rale koju potom koristimo u idućoj naredbi kako bismo izlistali sve dijeljene mape prisutne na tim računalima.

*

Gornjim štivom samo smo se pripremili za teledirigiranje Windows računalima, nismo time i ovladali! Na Internetu se množe razni savjeti na temu, o PShellu se objavljuju debele knjižurde..., no najbolja pomoć nam je doslovce „pred nosom" - lokalni PShell help. Ukoliko je Internet dostupan računalu na kojem radimo sa PShellom, taj help lako ažuriramo naredbom **update-help**. Potom rabimo **get-help** kako bismo za određenu naredbu dobili opće informacije i primjere uporabe, npr.:
get-help enter-pssession –examples