



**Microsoft** | Solution Accelerators

# PowerShell Support in SCAP 1.2

Michael Tan

Microsoft Corporation

November, 2011

# Topics

- Problems and Business Needs
- PowerShell Overview
- PowerShell configuration in SCAP 1.2
- Demo
- Q/A

# Problems and Business Needs

- Some new Microsoft security baselines are using PowerShell Cmdlets as the only method for security configuration automation (e.g. Exchange, SQL and soon to be Windows 8)
- Microsoft Common Engineering Criteria requires the use of PowerShell Cmdlets for all administrative tasks, including configuration management
- SCAP1.1 does not support scripting or more specifically PowerShell

# Windows PowerShell

New command-line shell and scripting language



```
C:\Windows\System32\WindowsPowerShell\v1.0\PowerShell.exe
PS C:\> Get-PSProvider

Name                Capabilities                Drives
----                -
Alias                ShouldProcess                {Alias}
Environment          ShouldProcess                {Env}
FileSystem           Filter, ShouldProcess       {C, D, WIN, Fav...}
Function             ShouldProcess                {Function}
Registry            ShouldProcess                {HKLM, HKCU}
Variable            ShouldProcess                {Variable}
Certificate          ShouldProcess                {cert}

PS C:\> Get-WmiObject Win32_ComputerSystem

Domain                : ntdev.corp.microsoft.com
Manufacturer          : Dell Inc.
Model                 : Inspiron 9300
Name                  : JPSVISTA1
PrimaryOwnerName     : jsnover
TotalPhysicalMemory  : 2146279424

PS C:\>
```

- As **interactive** and **composable** as BASH/KSH
- As **programmatic** as Perl/Python/Ruby
- As **production oriented** as AS400 CL/VMS DCL
- Allows access to data stores as **easy to access** as filesystem
- Consistent interface to access configurations

# Demo

# Usage in Microsoft

- Common Engineering Criteria
  - *Deliver a complete set of task-oriented [cmdlets](#) which cover: **Configuration (read/write)**, Operational verification tests, Lifecycle, Security, Diagnostics, and Data Management.*
- Broad, enthusiastic product adoption
  - Customer feedback and move to services
- Strong momentum and investment stream

# PowerShell Openness

- Used IEEE Std 1003.2-1992 Posix Shell as a starting point for the language and VMS DCL syntax for CLI
- Submitted to the DMTF SMASH committee to standardize CLI and scripting for ALL Oses and HW management
  - Converted from VMS to UNIX CLI syntax to increase probability of standardization
- PASH – Open Source PowerShell
- PowerShell Remoting Protocol [MS-PSRP] published as an Open Specification
  - [http://msdn.microsoft.com/en-us/library/dd357801\(PROT.10\).aspx](http://msdn.microsoft.com/en-us/library/dd357801(PROT.10).aspx)
- Language licensed under the Community Promise
  - <http://blogs.msdn.com/b/powershell/archive/2011/04/16/powershell-language-now-licensed-under-the-community-promise.aspx>

# Q/A



# PowerShell Support in SCAP 1.2

- OVAL Requirements
- Design Considerations
- Samples
- Demo

# Requirement Considerations

- Establishing a framework allows mixing of different check approaches
- Providing built in support for value parameterization
- Leveraging an industry proven model for interactions with XCCDF
- Requiring the simplest possible statements based upon the cmdlet needed, the input parameters, and expected output
- Leading to atomic checks that focus on single low-level configuration statements

# Design Considerations

- Build OVAL definitions using a constrained PowerShell interface instead of allowing arbitrary scripting
- The new PowerShell identifiers must be similar to the existing OVAL type identifiers
- Keep the initial design/schema simple and cover major scenarios
- Build the initial definition with future extension in mind
- Complete data model and structure covers both GET (security check/monitoring) and SET (deployment), not format dependent

# Demo

- Configuration Management through PowerShell

# Microsoft Proposal with sample (1)

User scenario: collect Exchange server anti-spam update configuration

## Proposed PowerShell Cmdlet

```
> Get-AntiSpamUpdates -Identity contoso-server | Select-Object -Property  
SpamSignatureUpdatesEnabled
```

## Proposed SCAP

<win-def:cmdlet\_object id="oval:sample:obj:14" version="1" comment="check the type of Microsoft Forefront Security for Exchange Server anti-spam updates that are retrieved.">

```
<modulename>Microsoft.Exchange.Configuration</modulename>  
<moduleid></moduleid>  
<moduleversion>1.0</moduleversion>  
<verb>Get</verb>  
<noun>AntiSpamUpdates</noun>  
<parameters>  
  <field name="Identity">contoso-server</field>  
</parameters>  
<select>  
  <field name="Property">SpamSignatureUpdatesEnabled</field>  
</select>  
</win-def:cmdlet_object>
```

## Data model UI

Module:

Module ID:

Version:

Verb:  Noun:

Parameters:

Name	Value
Name	Identity
...	...

Selection:

Name	Value
Property	SpamSignature...
...	...

# Benefits of proposed design

- Structuralized data/model
- Simple and precise format
- Scalable
- Extendable

# Microsoft Proposal with sample (2)

User Scenario: "Check all service hosts resource usage."

## Proposed PowerShell Cmdlet

```
> Get-Process -Name svchost | Select-Object -Property  
NonpagedSystemMemorySize, PagedSystemMemorySize, PeakPagedMemorySize, PeakWorkingSet
```

## Proposed SCAP

```
<win-def:cmdlet_object xmlns="http://oval.mitre.org/XMLSchema/oval-definitions-6#PowerShell"  
id="oval:sample:obj:2" version="1" comment="Check all service hosts resource usage.">  
  <modulename>Microsoft.PowerShell.Management</modulename>  
  <moduleversion>2.0</moduleversion>  
  <verb>Get</verb>  
  <noun>Process</noun>  
  <parameters>  
    <field name="Name">svchost</field>  
  </parameters>  
  <select>  
    <field name="Property">  
NonpagedSystemMemorySize, PagedSystemMemorySize, PeakPagedMemorySize, PeakWorkingSet</field>  
  </select>  
</win-def:cmdlet_object>
```

# SCM Prototype Demo



# Q/A

# **Microsoft<sup>®</sup>**

*Your potential. Our passion.<sup>™</sup>*

© 2007 Microsoft Corporation. All rights reserved. Microsoft, Windows, Windows Vista and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries.

The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation.

MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.