Microsoft | Solution Accelerators

Lessons/Tips of Creating PowerShell Configuration in OVAL

Michael Tan
Microsoft - Security & Compliance

Agenda

Introduction

Samples

Solutions

Data Model

Problems

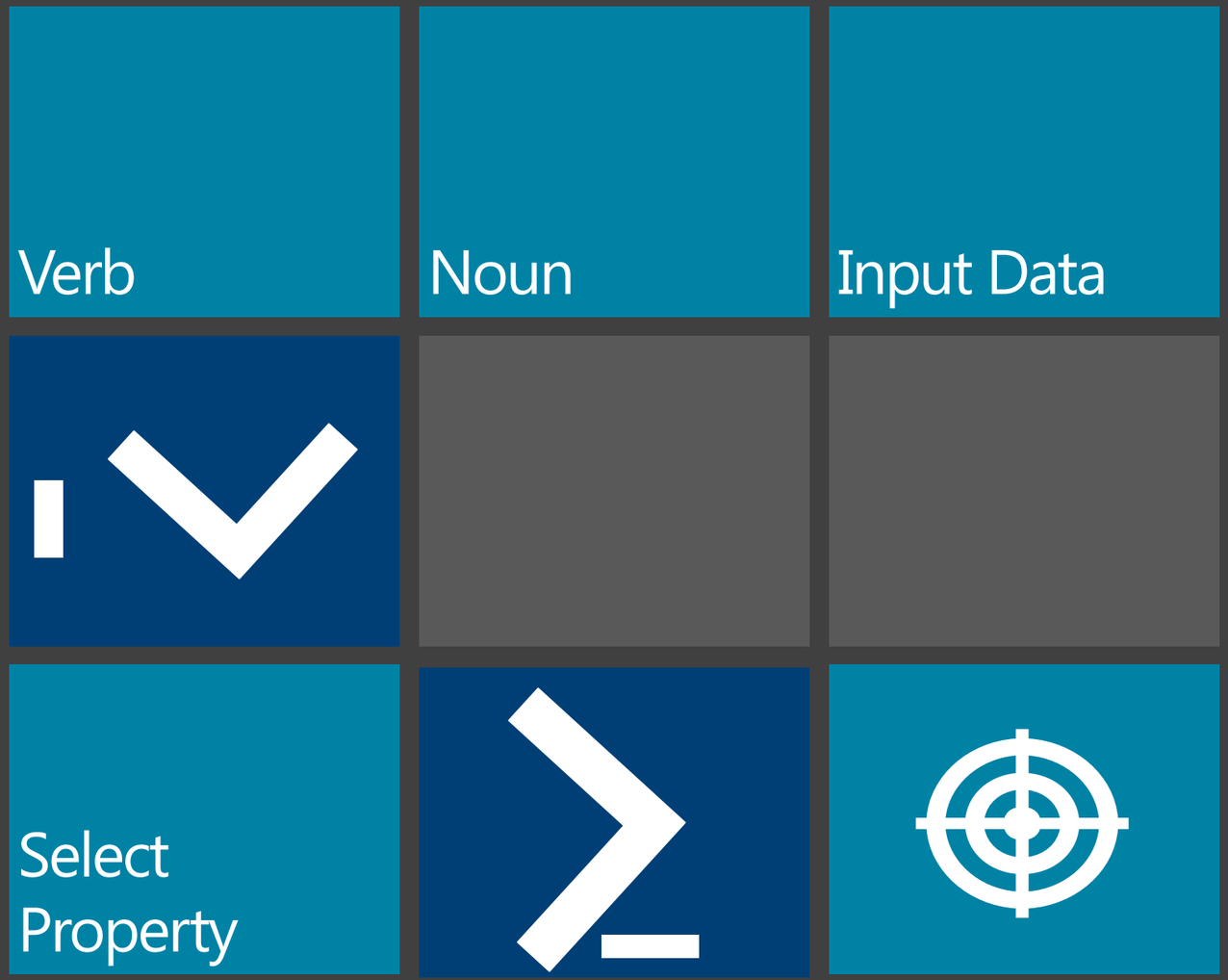
Demo

Introduction

- Identify Problem
- Community Discussion
- Presented 2011 SCAP Dev Day
- Positive Feedbacks
- Implementation
- In SCAP 1.2 Release in 2011

PowerShell Configuration Data Model

- Cmdlet as Interface
- Metadata to call
- Metadata to target
- Simple
- Flexible and Scale



Sample

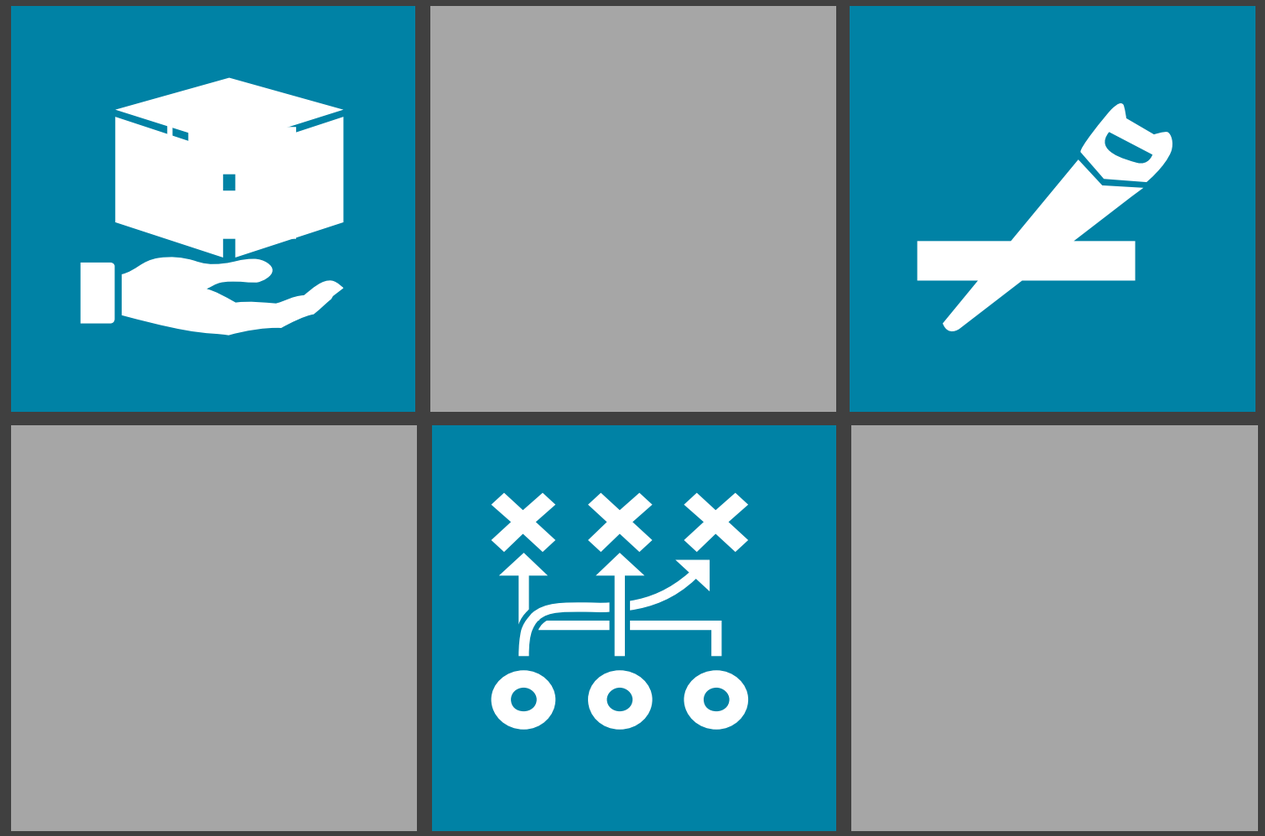
Exchange setting, Turn on Administrator Audit Logging

```
 <cmdlet_object id="oval:microsoft.com:obj:1">
<module_name>Microsoft.SolutionAccelerator.Baseline.Exchange</module_name>
    <module_id>{f1486d15-04a1-4782-82e9-981c2b986f4d}</module_id>
    <module_version datatype="version">1.0</module_version>
    <verb>Get</verb>
    <noun>ExchangeConfiguration</noun>
    <parameters datatype="record" operation="equals">
     <oval-def:field name="configtype">AdministratorAuditLogging</oval-def:field>
    </parameters>
    <select datatype="record">
     <oval-def:field name="property">SettingData</oval-def:field>
    </select>
   </cmdlet_object>
```

```
Get-ExchangeConfiguration  -configType ExecutionPolicy | Select-Object -Property SettingData
```

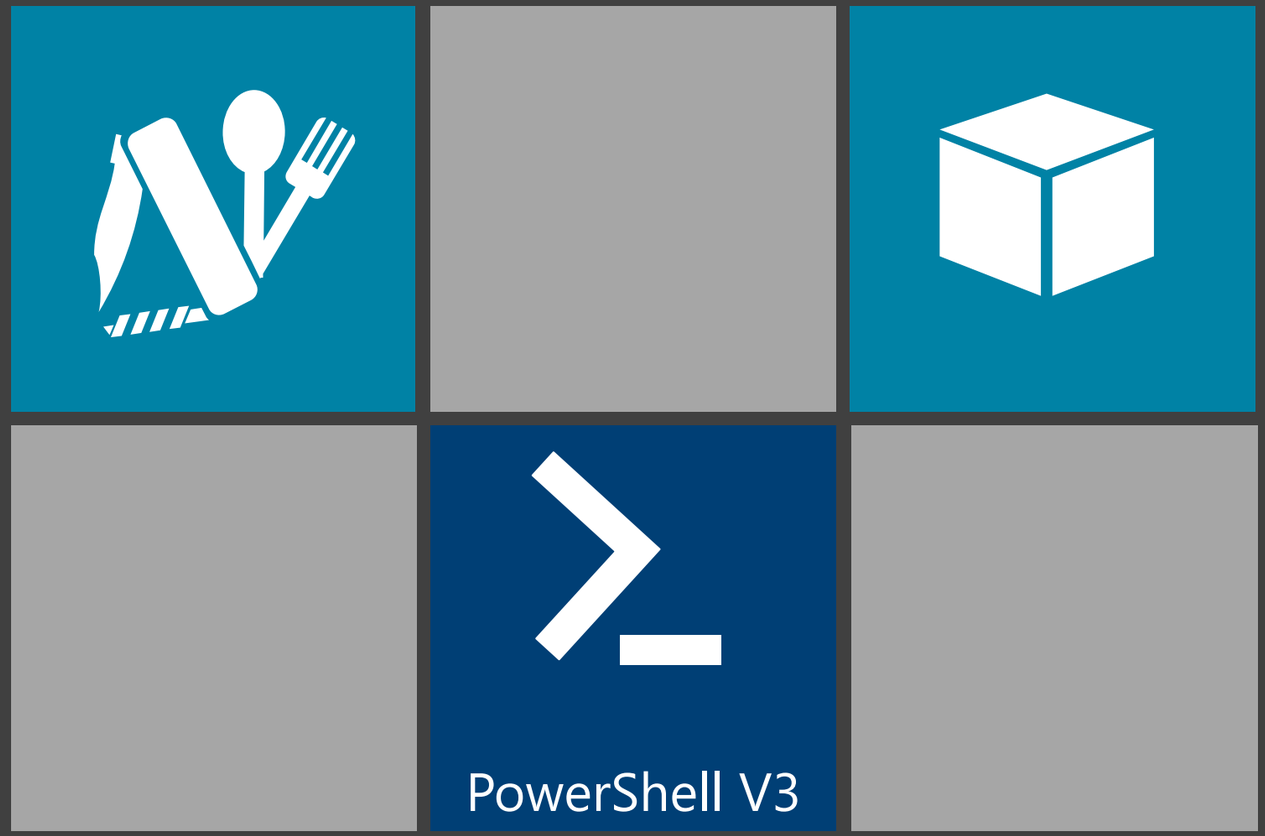
Problems

- Production Oriented Cmdlets? Reality Check…
- Compliance Requirements before product design or after?



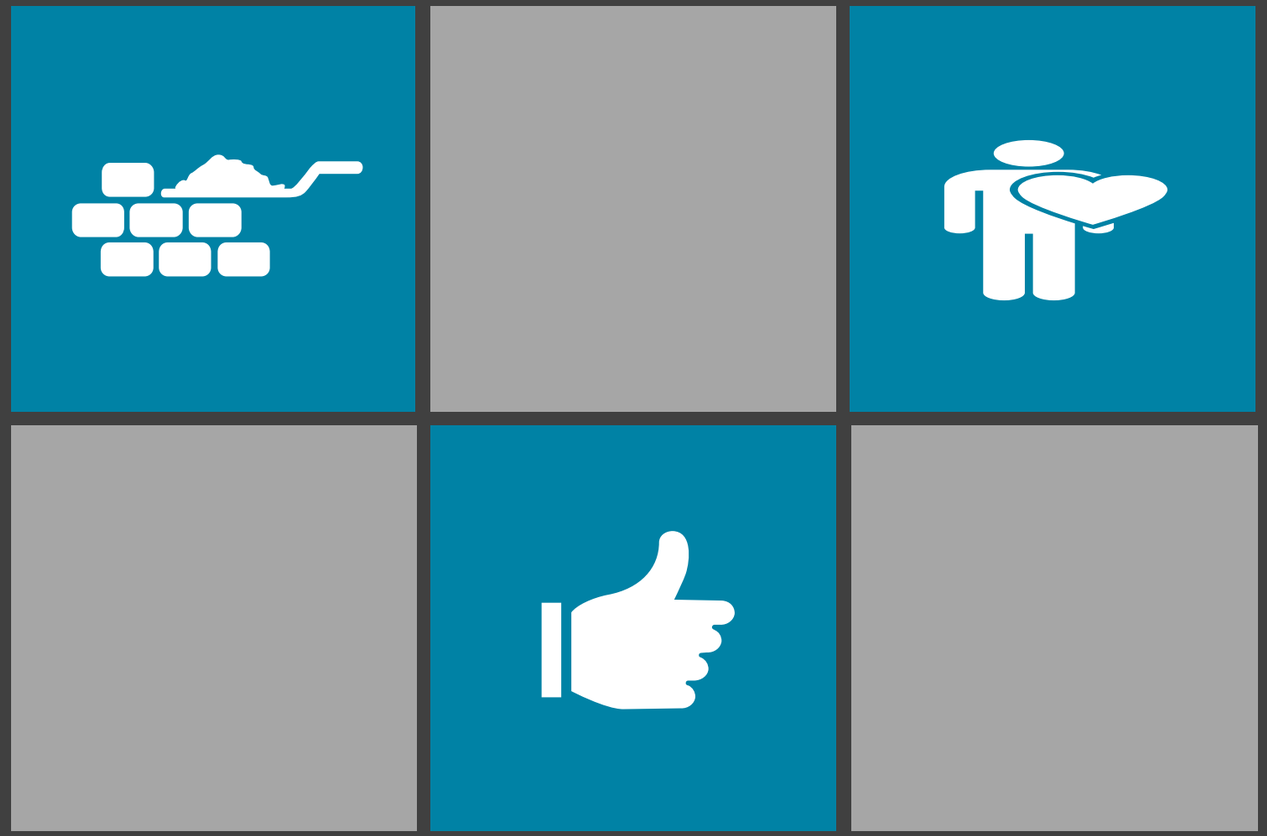
Solutions

- Write your own Cmdlets
- Simple to do in PowerShell V3
- PowerShell Script Function => Cmdlet

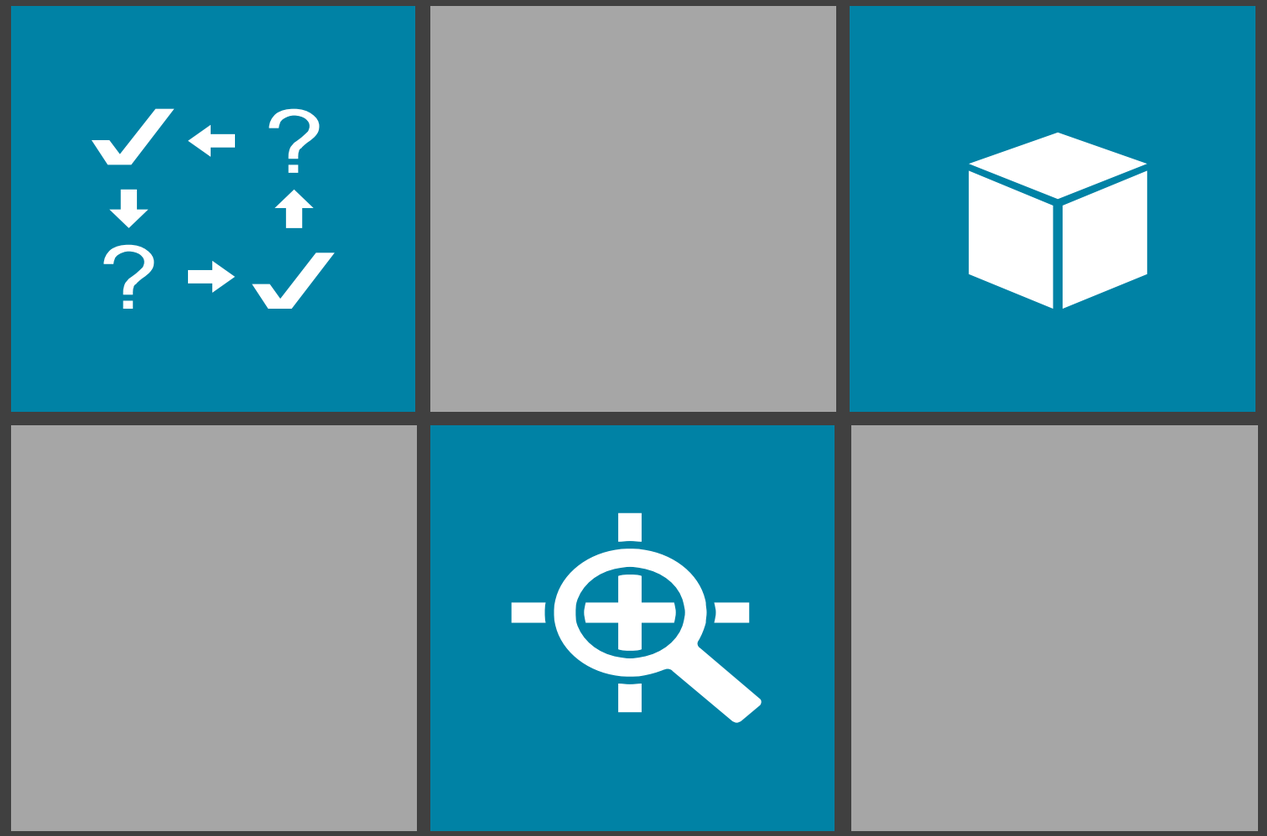


Demo

- Step by Step
- Line by Line
- Sample code attached



Q/A



Appendix, Working Sample Code

```
# save below into file SCAPDemoCmdlet.psd1

#
# Module manifest for module 'SCAPDemoCmdlet'
#
# Generated on: 7/9/2012
#

@{

# Script module or binary module file associated with this manifest
ModuleToProcess = 'SCAPDemoCmdlet.psm1'

# Version number of this module.
ModuleVersion = '1.0'

# ID used to uniquely identify this module
GUID = 'B9FE9A1F-FA59-48A2-9EAD-118B64758048'

# Author of this module
Author = 'SCAP Dev Day 2012, Demo'

# Company or vendor of this module
CompanyName = 'Contosole'

# Copyright statement for this module
Copyright = 'Copyright (c) 2012 Contosole'

# Description of the functionality provided by this module
Description = 'This is PowerShell script module for all security and compliance cmdlets'

# Minimum version of the Windows PowerShell engine required by this module
PowerShellVersion = ''

# Name of the Windows PowerShell host required by this module
PowerShellHostName = ''

# Minimum version of the Windows PowerShell host required by this module
PowerShellHostVersion = ''

# Minimum version of the .NET Framework required by this module
DotNetFrameworkVersion = ''

# Minimum version of the common language runtime (CLR) required by this module
CLRVersion = ''

# Processor architecture (None, X86, Amd64, IA64) required by this module
ProcessorArchitecture = ''

# Modules that must be imported into the global environment prior to importing this module
RequiredModules = @()

# Assemblies that must be loaded prior to importing this module
RequiredAssemblies = @()

# Script files (.ps1) that are run in the caller's environment prior to importing this module
ScriptsToProcess = @()

# Type files (.ps1xml) to be loaded when importing this module
TypesToProcess = @()

# Format files (.ps1xml) to be loaded when importing this module
FormatsToProcess = @()

# Modules to import as nested modules of the module specified in ModuleToProcess
NestedModules = @()

# Functions to export from this module
FunctionsToExport = 'Get-MyConfiguration'

# Cmdlets to export from this module
CmdletsToExport = 'Get-MyConfiguration'

# Variables to export from this module
VariablesToExport = '*'

# Aliases to export from this module
AliasesToExport = '*'

# List of all modules packaged with this module
ModuleList = @()

# List of all files packaged with this module
FileList = @()

# Private data to pass to the module specified in ModuleToProcess
PrivateData = ''

}
```

Appendix, Working Sample Code

Save below to file SCAPDemoCmdlet.psm1

```
function Get-MyConfiguration
{
  # custom cmdlet in PowerShell script is really a regular PowerShell function with CmdletBinding
  # CmdletBinding can have below attributes as options
  #   - SupportsShouldProcess=<Boolean>
  #   - ConfirmImpact=<String>
  #   - DefaultParameterSetName=<String>
  # they are not used in below example
  [CmdletBinding()]
  Param
  (
     # list of input parameters for the cmdlet
     [Parameter(
       Mandatory=$true,
       ValueFromPipeline=$true)]
     [string] $InputParam1,

     [Parameter(
       Mandatory=$false,
       ValueFromPipeline=$false)]
     [int] $InputParam2
  )
  Process
  {
    # here is the main process to collect data and make business logics to build MyConfiguration
    # TODO: replace below with your code

    # return object uses a dummy string but it be your object
    $o = "dummy" | Select-Object -Property dummyProperty
    # add output properties here, they are all note-properties
    $o = Add-Member -InputObject $o -MemberType NoteProperty -Name OutputProperty1 -Value "output property 1 value" -PassThru
    $o = Add-Member -InputObject $o -MemberType NoteProperty -Name OutputProperty2 -Value "output property 2 value" -PassThru
    return $o
  }
}
```