

# PowerShell Security - All you need to know

David das Neves  
Premier Field Engineer



Microsoft



SAPIEN  
Technologies, Inc.

# David das Neves

## Premier Field Engineer

Twitter: @david\_das\_neves

LinkedIn: DaviddasNeves

Blog: <http://aka.ms/ddnblog>

E-Mail: David.dasNeves@Microsoft.com



# Introduction

# Why PowerShell Security matters

The screenshot shows a blog post from the Symantec Official Blog. The title of the post is "PowerShell threats surge: 95.4 percent of analyzed scripts were malicious". The author is Candid Wueest, a Symantec employee. The post was created on 08 Dec 2016 and has 0 comments. It includes social sharing buttons for Google+, LinkedIn, Twitter, and Facebook, with counts of 10, 305, 1, and 1 respectively. A small thumbnail image in the bottom left corner shows a globe with network connections and the text "THE INCREASED USE OF POWERSHELL IN ATTACKS".

Cyber Security ← Security Response

Symantec Official Blog

## PowerShell threats surge: 95.4 percent of analyzed scripts were malicious

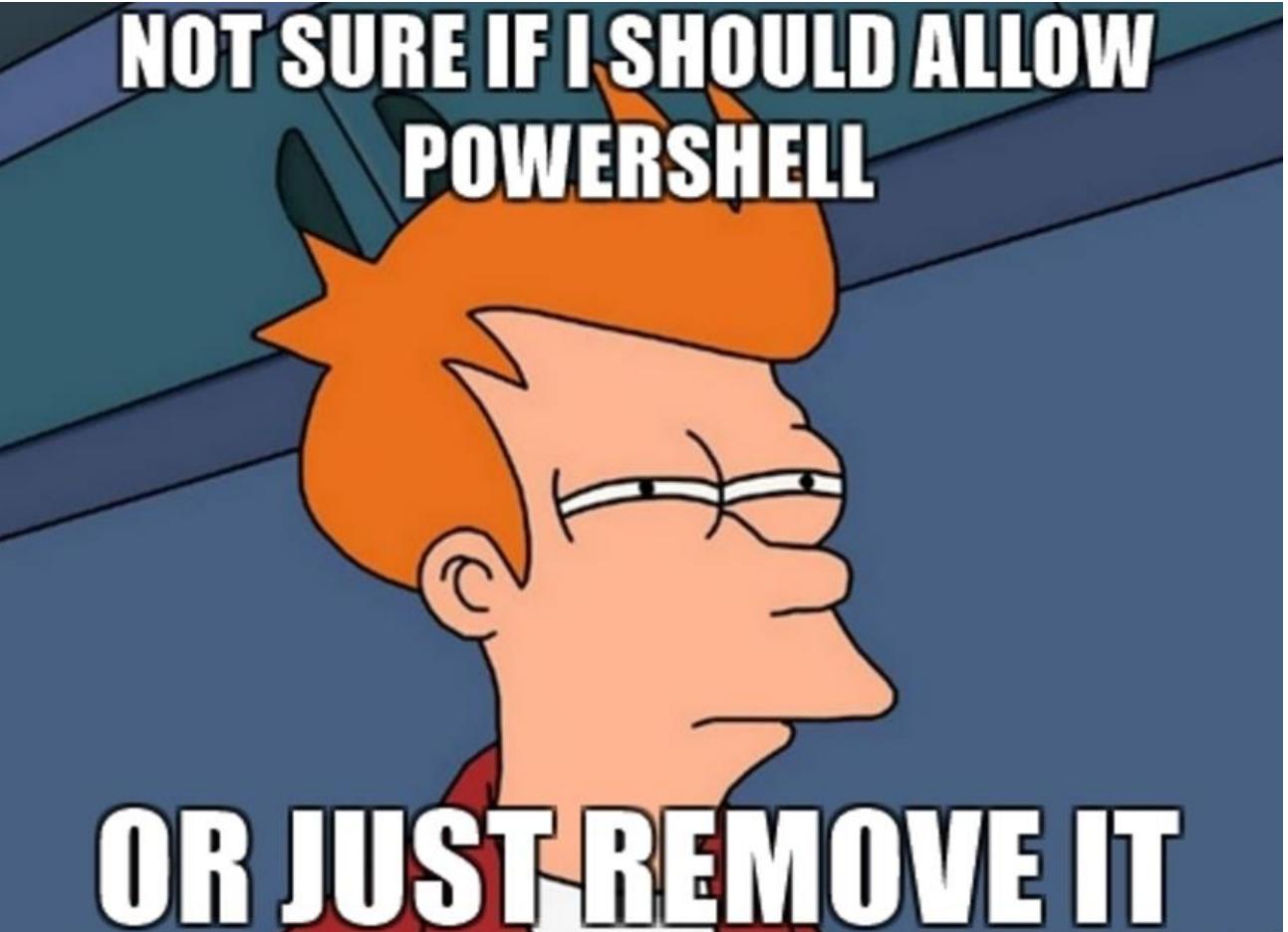
Symantec analyzed 111 threat families that use PowerShell, finding that they leverage the framework to download payloads and traverse through networks.

By: Candid Wueest SYMANTEC EMPLOYEE ACCREDITED

Created 08 Dec 2016 0 Comments 简体中文, 日本語

g+ 10 in 305 tw 1 fb Like 1

THE INCREASED USE OF POWERSHELL IN ATTACKS



**NOT SURE IF I SHOULD ALLOW  
POWERSHELL**

**OR JUST REMOVE IT**

# Some Notes from the Field

"We don't know PowerShell Security – therefore we shut it down completely."

"InfoSec will not let us turn on PowerShell remoting."

"Our last audit said that PowerShell needs to be locked down on all servers."

"It is unsecure – you can read it in the news!"

"The CIO went to a security conference and then banned PowerShell from the environment."

"The BSI\* recommended to do so."

# Top 10 reasons why attackers use PowerShell

1. Installed by default

2. Can be executed fileless

3. Generates few traces by default

4. Has remote access capabilities by default

5. Easy to obfuscate

# Top 10 reasons why attackers use PowerShell

6. Defenders often overlook it when hardening
7. Can bypass application-whitelisting
8. Many sandboxes do not handle script-based malware well.
9. Has growing community with ready available scripts.
10. Many system administrators use and trust the framework

# A Comparison of Shell and Scripting Language Security

Engine	Event Logging	Transcription	Dynamic Evaluation Logging	Encrypted Logging	Application Whitelisting	Antimalware Integration	Local Sandboxing	Remote Sandboxing	Untrusted Input Tracking
Bash	No**	No*	No	No	Yes	No	No*	Yes	No
CMD / BAT	No	No	No	No	Yes	No	No	No	No
Jscript	No	No	No	No	Yes	Yes	No	No	No
LUA	No	No	No	No	No	No	No*	Yes	Yes
Perl	No	No	No	No	No	No	No*	Yes	Yes
PHP	No	No	No	No	No	No	No*	Yes	Yes
PowerShell	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No**
Python	No	No	No	No	No	No	No	No	No**
Ruby	No	No	No	No	No	No	No**	No**	Yes
sh	No**	No*	No	No	No	No	No*	Yes	No
T-SQL	Yes	Yes	Yes	No	No	No	No**	No**	No
VBScript	No	No	No	No	Yes	Yes	No	No	No
zsh	No**	No*	No	No	No	No	No*	Yes	No

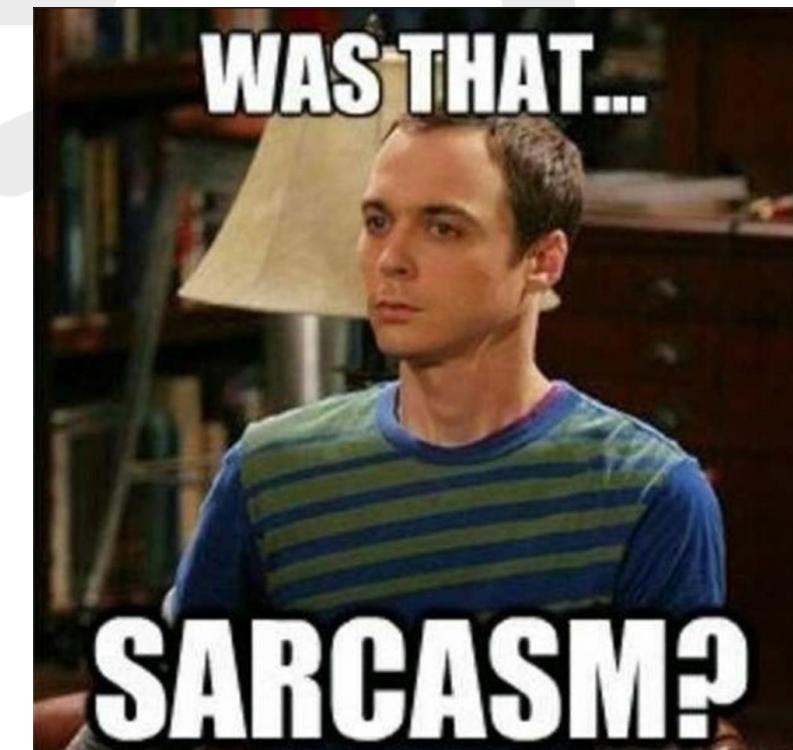
\* Feature exists, but cannot enforce by policy  
\*\* Experiments exist

# PowerShell Security

What matters.

# ExecutionPolicy is a ~~parameter~~

Oh wait..



# What is most important?



# PowerShell is more than PowerShell.exe

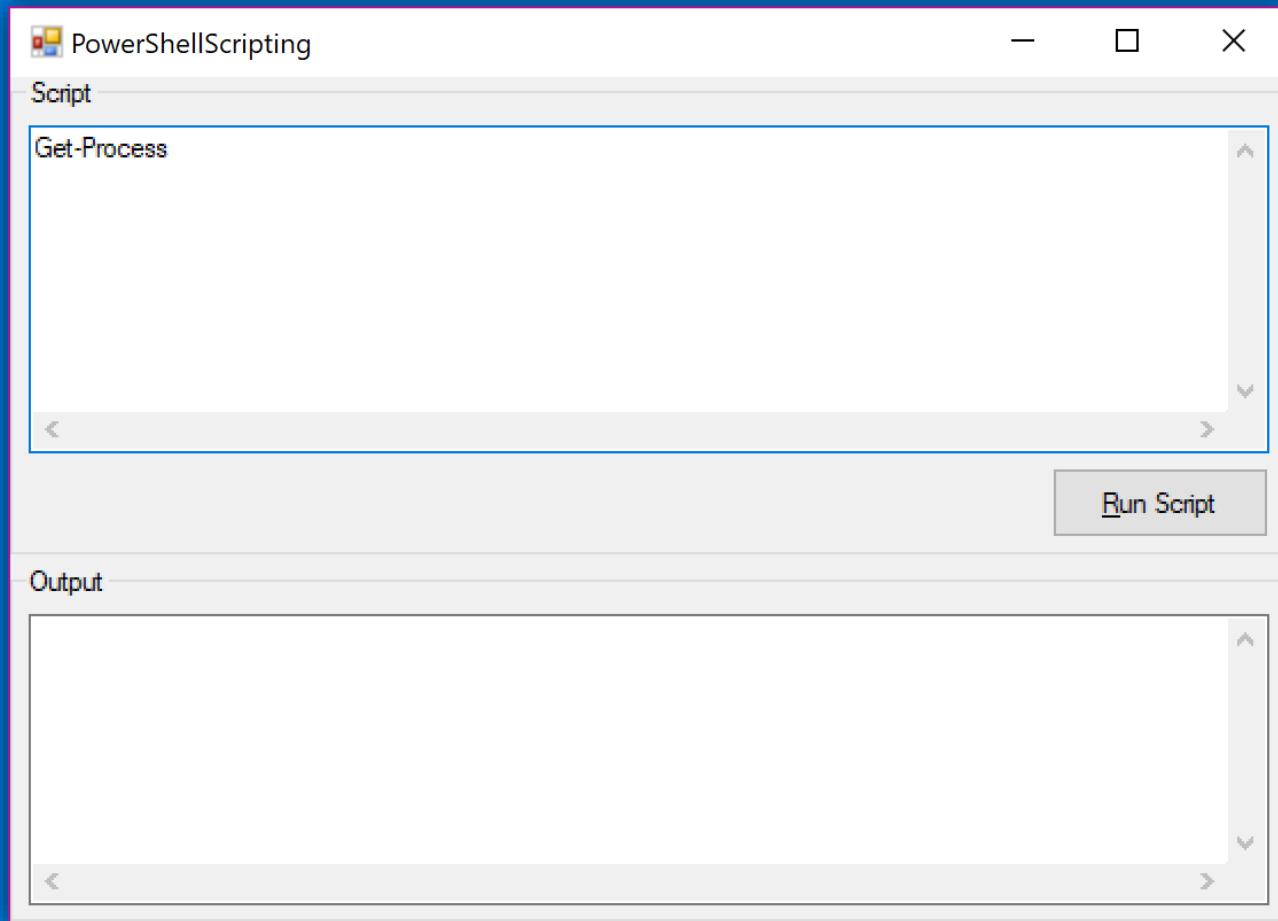
## System.Management.Automation Namespace

Updated: April 27, 2016

Applies To: Windows PowerShell

The System.Management.Automation namespace is the root namespace for Windows PowerShell. It contains classes, enumerations, and interfaces. For example, cmdlet developers will use the [Cmdlet](#) and [PSCmdlet](#) base classes to implement custom cmdlet classes, and host application developers will use the [PowerShell](#) class to create runspaces and invoke commands.

# Demo – C#Power



# PowerShell is more than PowerShell.exe

The screenshot shows a Windows Features dialog box overlaid on a background of a PowerShell session monitoring tool. The dialog box is titled "Turn Windows features on or off". It lists several features with checkboxes:

- Simple TCPIP services (i.e. echo, daytime etc)
- SMB 1.0/CIFS File Sharing Support
- Telnet Client
- TFTP Client
- Unified Write Filter
- Windows Identity Foundation 3.5
- Windows PowerShell 2.0
  - Windows PowerShell 2.0 Engine
- Windows Process Activation Service
- Windows TIFF IFilter
- Work Folders Client
- XPS Services
- XPS Viewer

At the bottom of the dialog box are "OK" and "Cancel" buttons.

In the background, a PowerShell session is visible, showing a table of operations and a list of files under "Windows Features".

# Remoting



# Some companies shut down PowerShell, but ...

will leave any one or more of these remote management ports open:

Remote Desktop  
Protocol (RDP)

Remote WMI access  
over RPC, clear text  
by default, random  
ports

Remote event log  
management

Remote service  
management

SMB file share  
access

PSEXEC

# PowerShell Remoting

always  
encrypted

- single port
  - 5985 (http)
  - 5986 (https)
  - With certificate

In a domain only  
members of the  
Administrators  
group have the  
ability to  
remote.

Advanced  
logging  
possibilities

# PowerShell Remoting – Bottom Line

The improvements in **WMF 5.0** make PowerShell the worst tool of choice for a hacker when you enable script block logging and system-wide transcription.

Hackers will leave **fingerprints everywhere**, unlike popular CMD utilities.

**For this reason, PowerShell should be the *only* tool you allow for remote administration.**



# Logging

Event Properties - Event 4688, Microsoft Windows security auditing.

A new process has been created.

Subject:

Security ID:	[REDACTED]-DC\ [REDACTED]
Account Name:	[REDACTED]
Account Domain:	[REDACTED]
Logon ID:	0x46068

Process Information:

New Process ID:	0x210
New Process Name:	C:\Windows\System32\WindowsPowerShell\v1.0
\powershell.exe	
Token Elevation Type:	TokenElevationTypeDefault (1)
Creator Process ID:	0x4e0

Process Command Line: "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -w hidden -enc cGFyYW0gKCRDb21wdXRlck5hbWUgPSAiLilsICRGaWxIUGF0aCA9ICluXEFwcGxpY2F0aW9uc0lu dmVudG9yeS5jc3YiKQ0KDQpnZXQtd21pb2JqZWN0IC1xdWVySAiU0VMRUNUICogRJPTSBXa W4zMl9Qcm9kdWN0liAtY29tcHV0ZXJuYW1ICRDb21wdXRlck5hbWUgfCANCnNvcnQtb2JqZW N0IFZlbnRvcIB8IA0Kc2VsZWN0LW9iamVjdCBQU0NvbXB1dGVyTmFtZSzWZw5kb3lsTmFtZSzW ZXJzaW9uLENhcHRpb24sRGVzY3JpcHRpb24sSW5zdGFsbERhdGUssW5zdGFsbExvY2F0aW9uLEI uc3RhbgxTb3VY2UsUGFja2FnZU5hbWUgfA0KZXhwb3J0LWNzdiAtcGF0aCAkRmlsZVBhdGggL WFwcGVuZCA=

Token Elevation Type indicates the type of token that was assigned to the new process in accordance with User Account Control policy.

# Logging

Module  
Logging

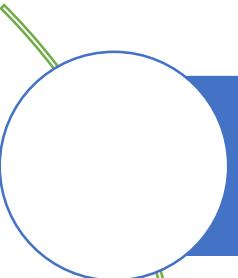
Transcription  
Logging

ScriptBlock  
Logging

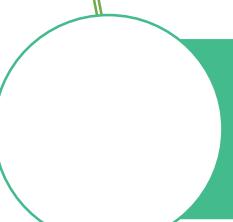


PowerShell Conference  
Singapore 2017

# Module Logging



records pipeline execution details, including variable initialization and command invocations.

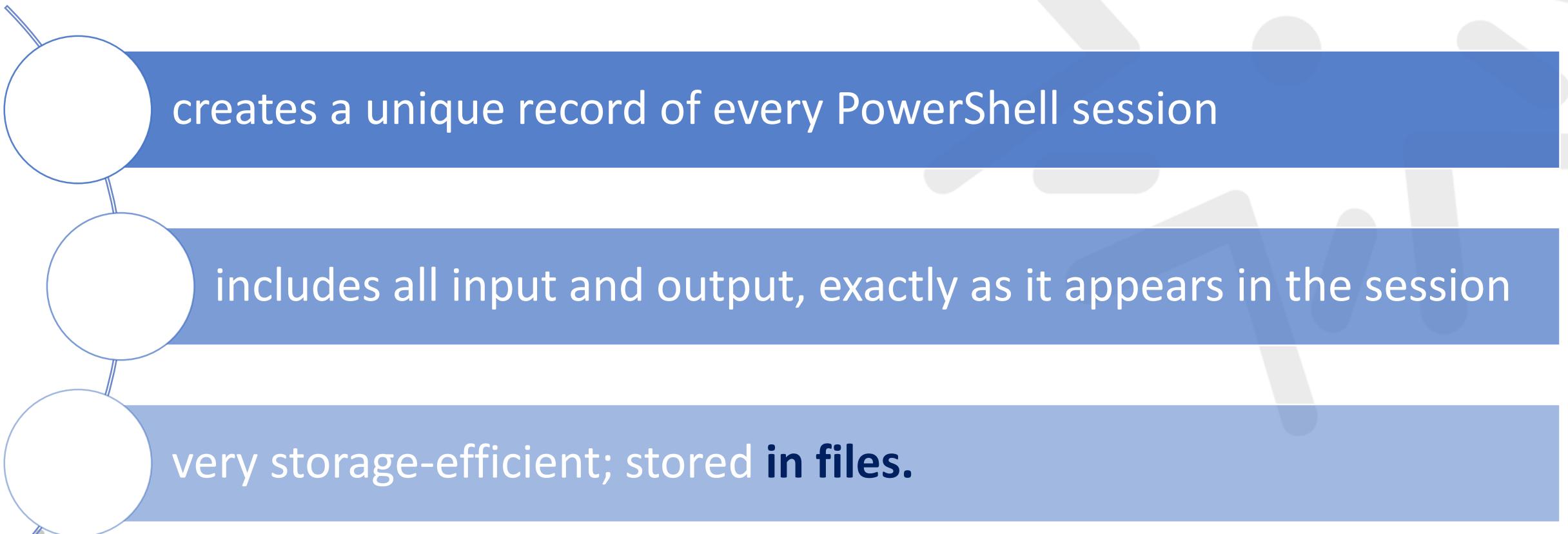


Available since PowerShell v3



events are written to Event ID **EID 4103**

# Transcription



creates a unique record of every PowerShell session

includes all input and output, exactly as it appears in the session

very storage-efficient; stored **in files**.

# Script Block Logging



logging records blocks of code as they are executed by the PowerShell engine

capturing the full contents of code executed by an attacker, including scripts + commands

also records **de-obfuscated code** logging events are recorded in Event ID  
**EID 4104**

# Logging – Best Practice

In environments with PowerShell 5.0, organizations should consider, at a minimum, aggregating and monitoring suspicious script block logging events, EID 4104 with level “warning”, in a SIEM or other log monitoring tool.

... , attacker commands and code execution.

# Configuration GPOs / MDM

The screenshot shows the 'Windows PowerShell' policy setting in the Group Policy Management Editor. The 'Setting' section includes options like 'Turn on Module Logging', 'Turn on PowerShell Script Block Logging' (which is selected), 'Turn on Script Execution', 'Turn on PowerShell Transcription', and 'Set the default source path for Update-History'. The 'Comment' field indicates the setting is enabled. The 'Supported on:' field lists supported operating systems. The 'Options' section contains a checked checkbox for 'Log script block invocation start / stop events'.

**Windows PowerShell**

**Turn on PowerShell Script Block Logging**

[Edit policy setting](#)

**Requirements:**  
At least Microsoft Windows 7 or Windows Server 2008 family

**Description:**

This policy setting enables logging of all PowerShell script input to the Microsoft-Windows-PowerShell/Operational event log. If you enable this policy setting, Windows PowerShell will log the processing of commands, script blocks, functions, and scripts - whether invoked interactively, or through automation.

If you disable this policy setting, logging of PowerShell script input is disabled.

If you enable the Script Block Invocation Logging, PowerShell additionally logs events when invocation of a command, script block, function, or script

**Setting**

- Turn on Module Logging
- Turn on PowerShell Script Block Logging**
- Turn on Script Execution
- Turn on PowerShell Transcription
- Set the default source path for Update-History

**Comment:**  
 Not Configured       Enabled       Disabled

**Supported on:**

Log script block invocation start / stop events

# ConstrainedLanguage Applocker Device Guard



# Constrained Language Mode

allows only  
core  
PowerShell  
functionality

prevents execution of the extended language  
features often used by offensive PowerShell tools

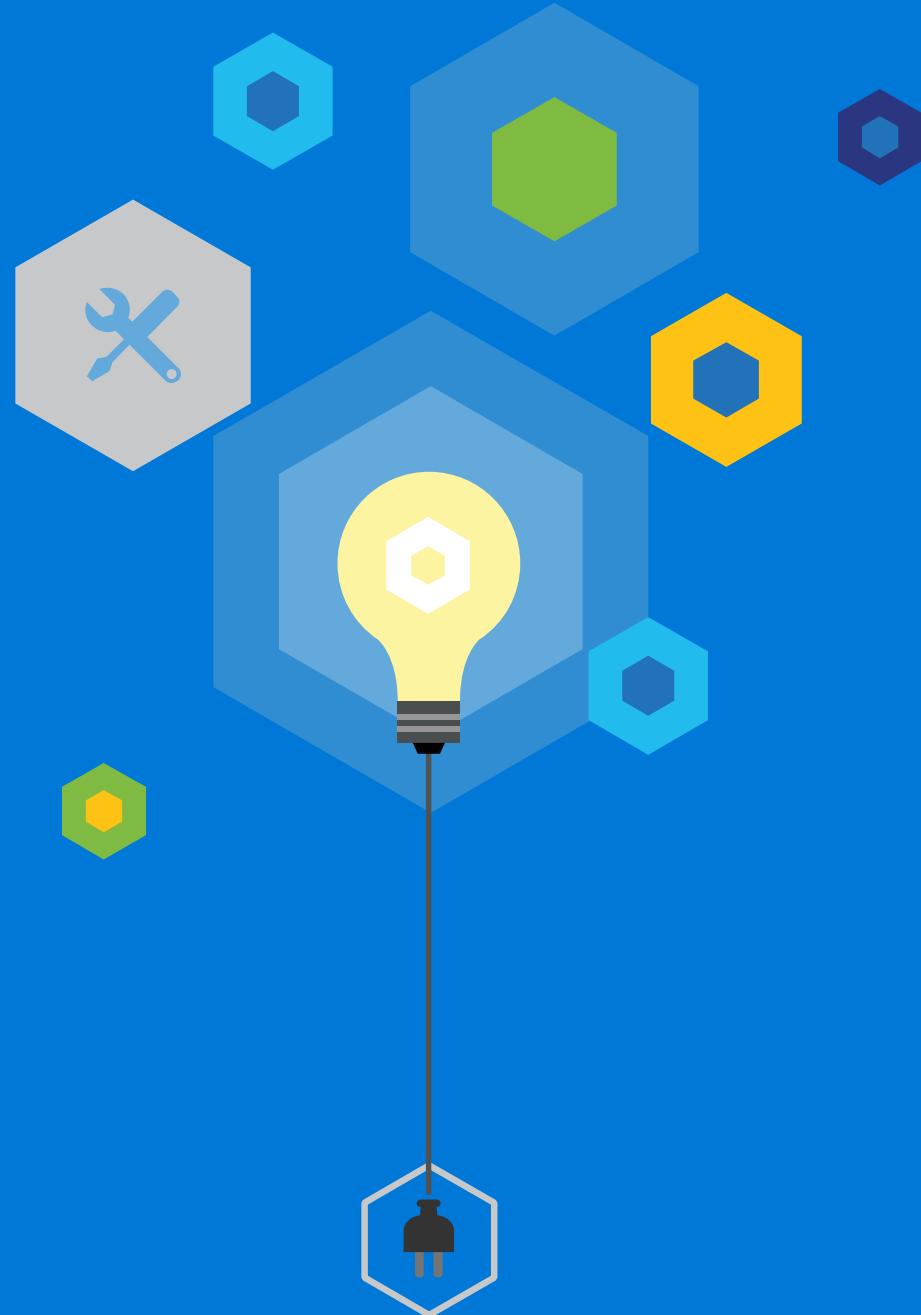
direct .NET  
scripting

invocation of  
Win32 APIs via  
the Add-Type  
cmdlet

interaction  
with COM  
objects



# Demo - ConstrainedLanguage



# PowerShell v5 changes everything

```
PS C:\Users\dadasnev> $PSVersionTable
```

Name	Value
---	----
PSVersion	5.1.16288.1
PSEdition	Desktop
PSCompatibleVersions	{1.0, 2.0, 3.0, 4.0...}
BuildVersion	10.0.16288.1
CLRVersion	4.0.30319.42000
WSManStackVersion	3.0
PSRemotingProtocolVersion	2.3
SerializationVersion	1.1.0.1



# POWERSHELL VS

T H E   F O R C E   A W A K E N S

May the enforce be with you

PowerShell v5

Constrained Language

AppLocker in Allow  
Mode

Device Guard with  
enforced UMCI



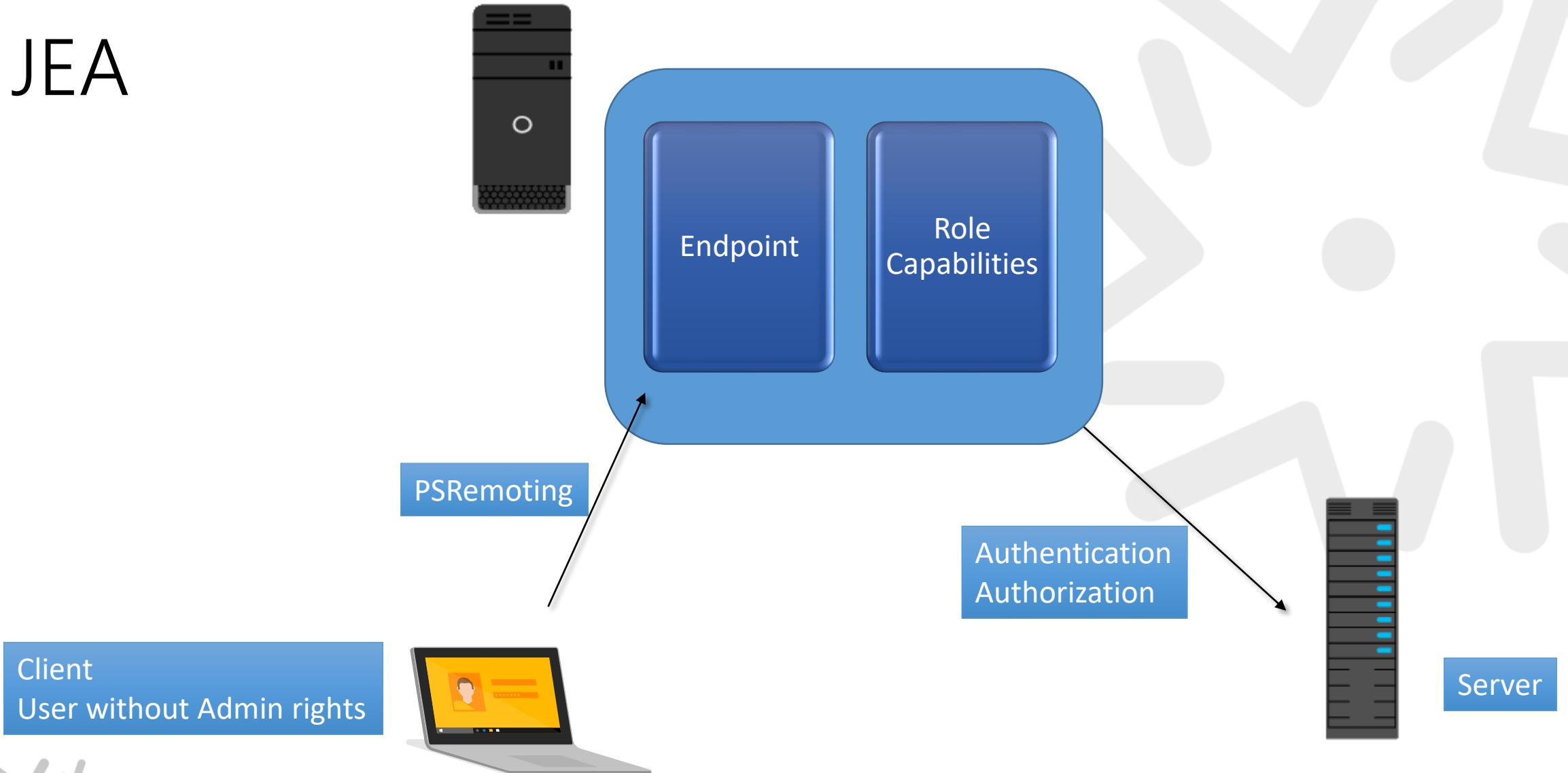
# AppLocker / DeviceGuard + PowerShell v5

Will need internal  
repositories and new  
processes

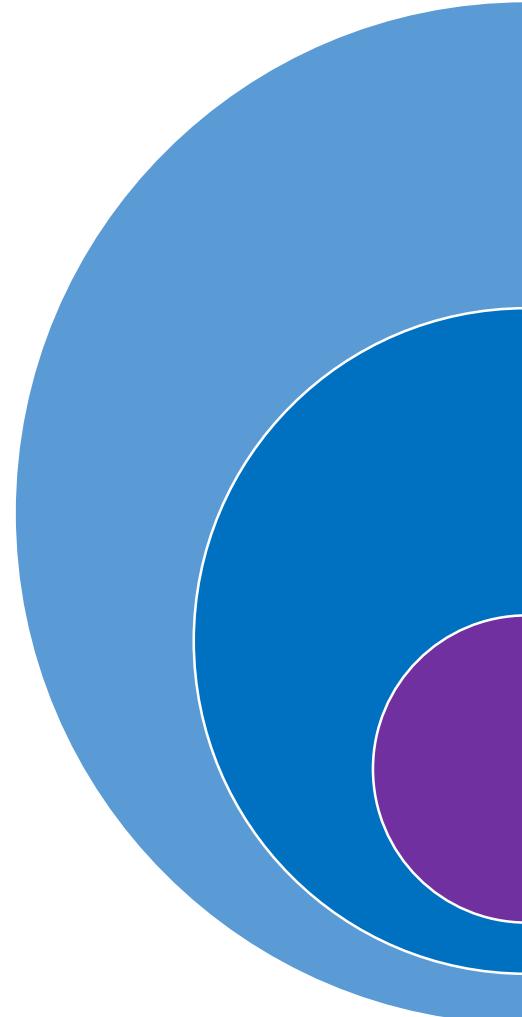


# Just Enough Administration

# JEA



JEA



Minimizing Attack vector  
by reducing

Capabilities

For a defined time



PowerShell Conference  
Singapore 2017

Windows 10 Features  
Server 2016 Features

# Windows 10 / Server 2016 brings some more

Credential  
Guard

Exploit  
Guard

AMSI

Windows  
Defender  
Application  
Guard

Windows  
Defender  
Advanced  
Threat  
Protection

Hello for  
Business

# Future

Enforced Constrained  
Language is a very  
secure hardening



**Matt Graeber**  
@mattifestation

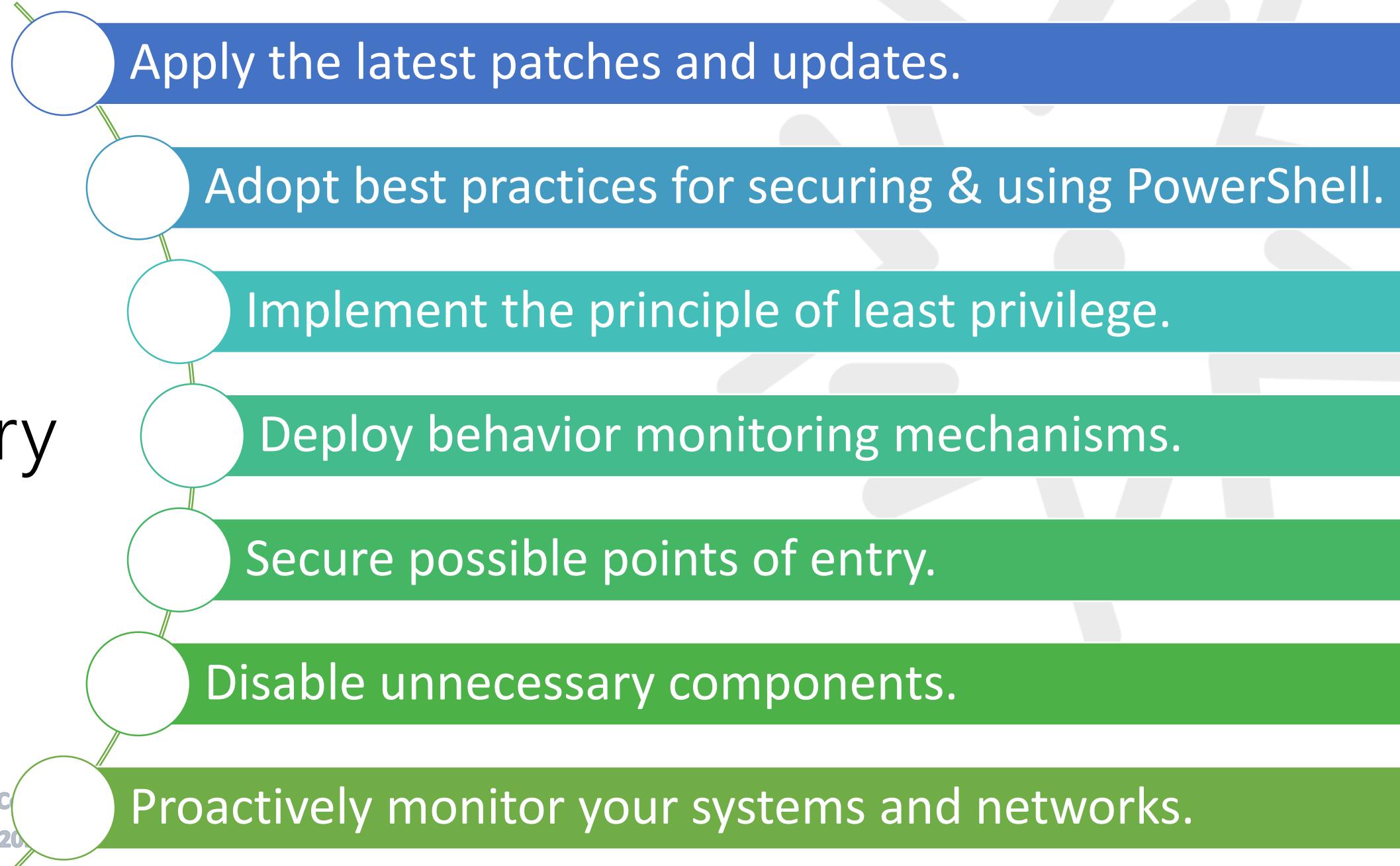
Following

The security transparency is so good in PowerShell that I'm investing in other languages/frameworks w/ less/no security insight.

4:37 PM - 8 Sep 2017

PowerShell is becoming  
unattractive for pentesting  
companies and hackers

## Summary



# Links

<https://blogs.msdn.microsoft.com/daviddasneves/2017/05/25/powershell-security-at-enterprise-customers/>

## Don't Forget!

- Fill in your survey in Mobile app – it's how we do better!
- Don't lose you badge! You need it for the Social Events
- Grab the Speakers for a chat – they all have time for you!
- Let everyone know what they are missing on Social Media  
#PowerShell  
#PSConfAsia

Photos of Marina Bay Credit: Sebastian Szumigalski