

Unlock the Power of PowerShell across your organization

Automate. Delegate. Relax.

Heiko Brenn  @heikobrenn

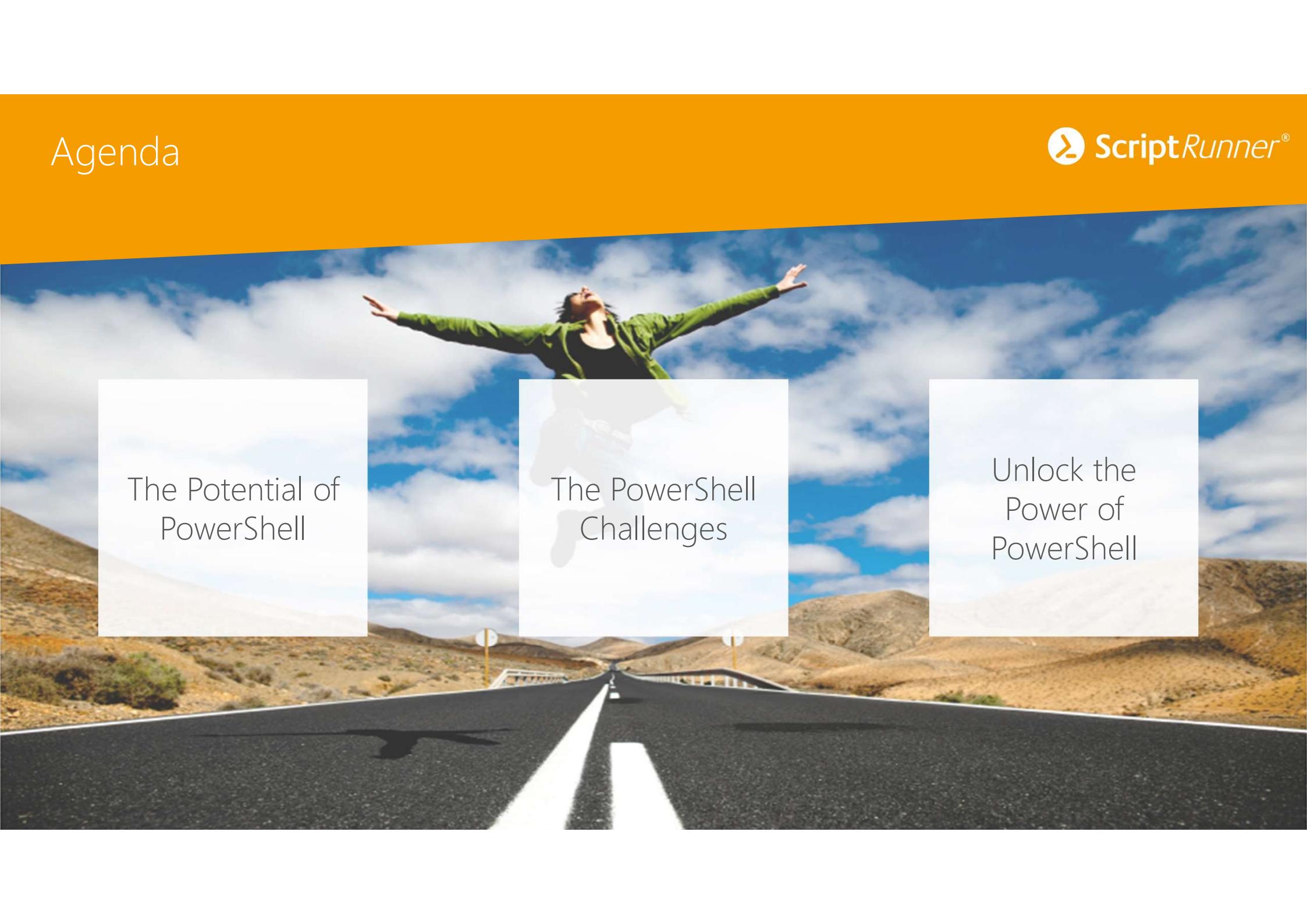
www.scriptrunner.com



Thanks to the Cloud8 sponsors



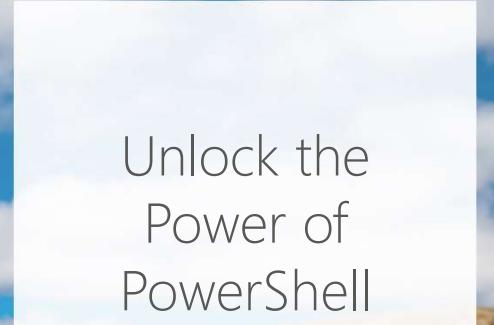
Agenda

A photograph of a person standing on a road in a desert landscape, with their arms outstretched upwards. The background shows a clear blue sky with scattered clouds and distant hills.

The Potential of
PowerShell

A silhouette of a person carrying a briefcase, walking towards the camera. This image is positioned over the central white box.

The PowerShell
Challenges

A silhouette of a person with arms outstretched, standing in a desert landscape under a blue sky with clouds. This image is positioned over the rightmost white box.

Unlock the
Power of
PowerShell



www.scriptrunner.com

The screenshot shows a detailed PowerShell reference guide. Key sections include:

- COMMON OPERATORS**: Includes operators like +, -eq, -ne, -gt, -lt, -ge, -le, -in, -not, -and, -or, -noteq, -notin, -notand, -notor.
- DATA TYPES**: Describes objects, arrays, hash tables, and more.
- LOOPS**: Examples of for, while, do, foreach, and switch loops.
- COMMANDS**: A comprehensive list of cmdlets, such as Get-Service, Get-Member, Select-Object, Sort-Object, and many others.



linkedin.com/in/HeikoBrenn

github.com/HeikoBrenn

+49-162-4879156

heiko.brenn@scriptrunner.com

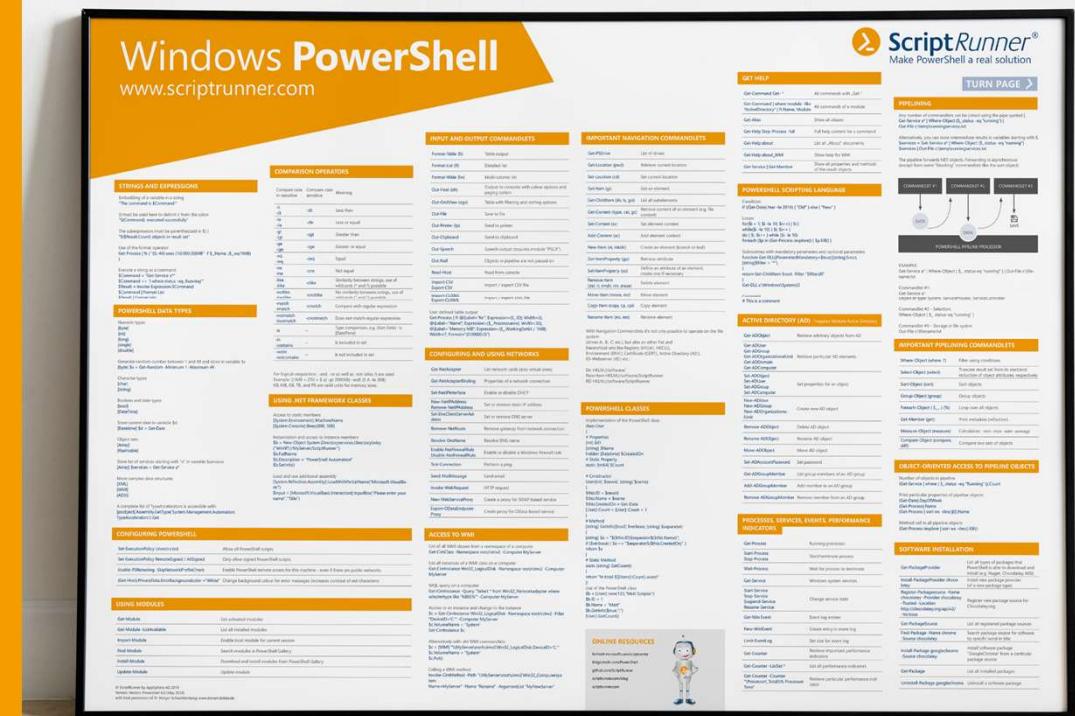


ScriptRunner®

The #1 for PowerShell Management

Get your free PowerShell poster

lp.scriptrunner.com/en/powershell-poster



The Potential of PowerShell



The Potential of PowerShell



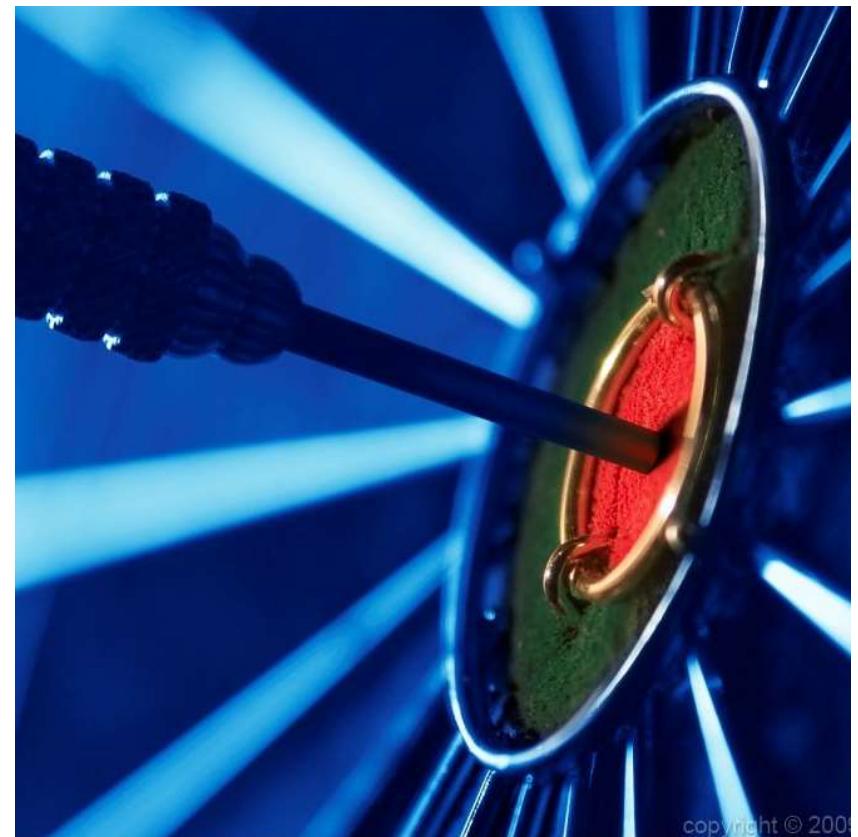
Very mature command-line shell
and scripting language for IT automation

Great framework to automate
recurring tasks consistently

Simplifies administration of your
entire IT infrastructure

Strong commitment from Microsoft
and many other vendors

Very vital worldwide community



copyright © 2005

Cross-platform PowerShell



PowerShell AnyWhere

Windows PowerShell
since 2006

PowerShell 7
since 2016

.NET Framework

.NET Core

Windows

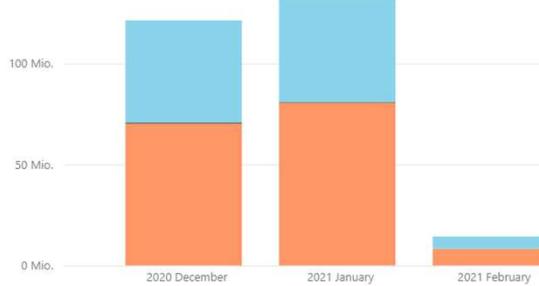
Linux

macOS

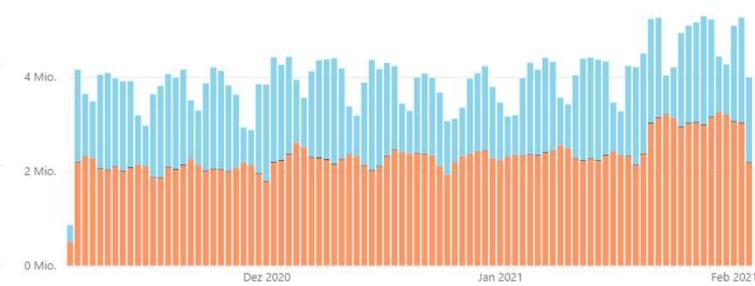
Cross-platform PowerShell



Monthly Usage by OS (90 Day)
userAgent ● Linux ● macOS ● Windows

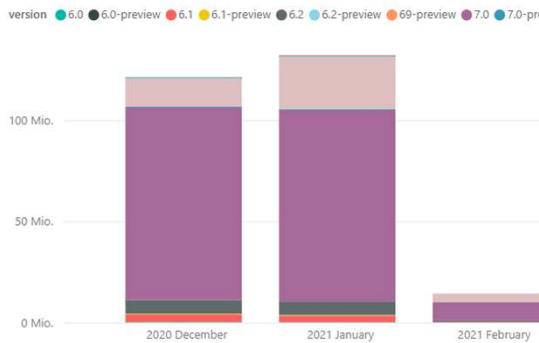


Daily Usage by OS (30 Day)
userAgent ● FreeBSD ● Linux ● macOS ● Windows

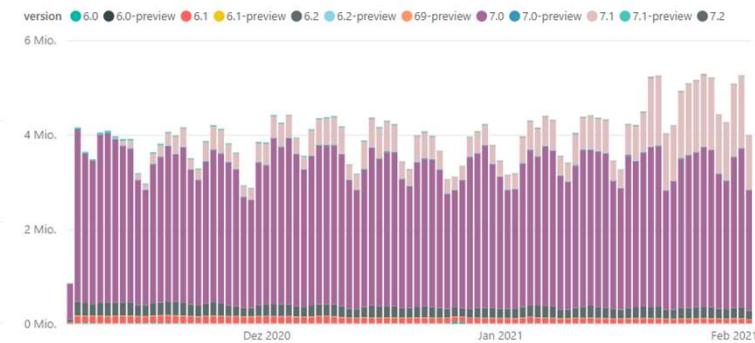


03.02.2021 00:00:00

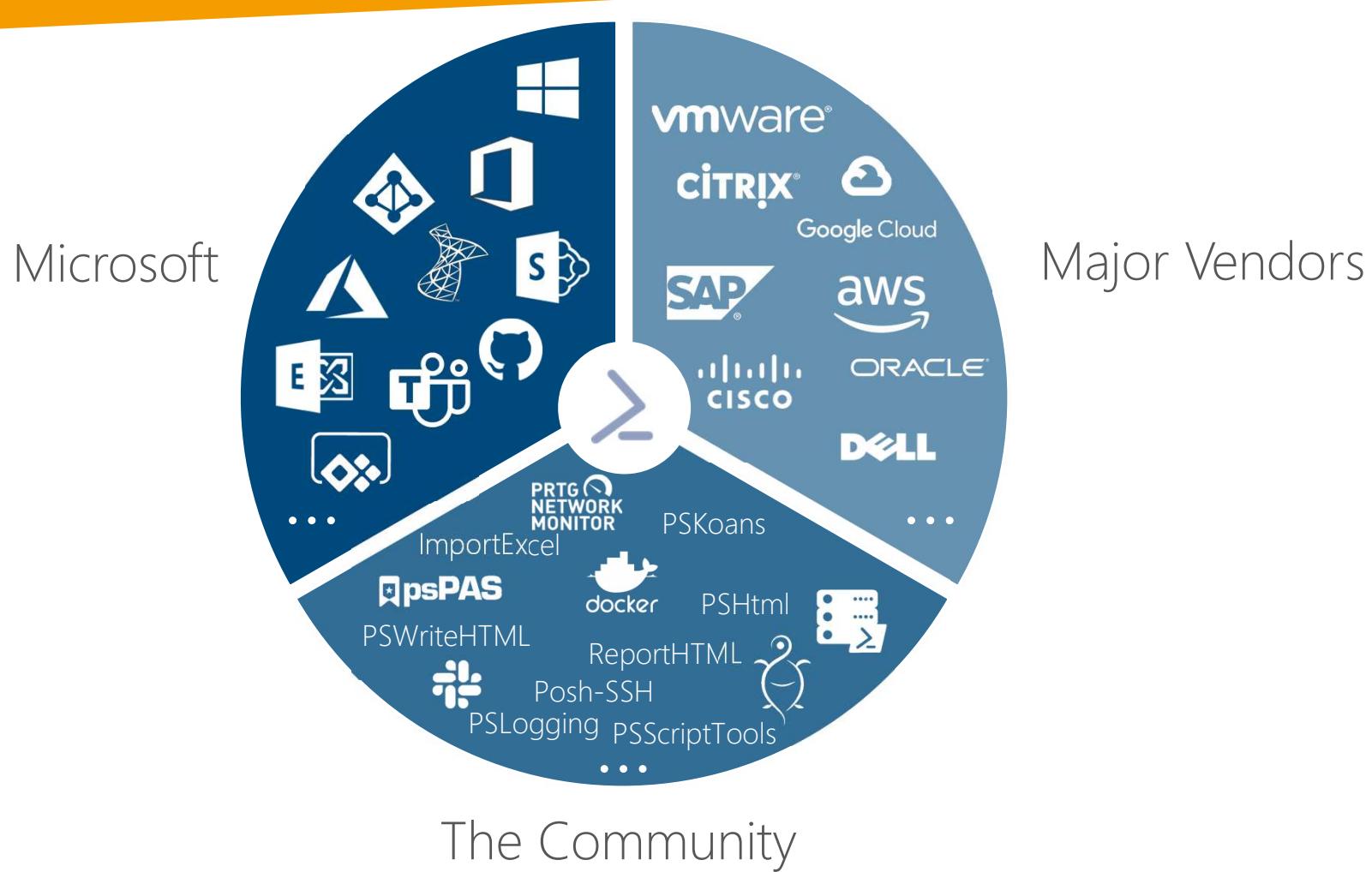
Monthly Usage by Version (90 Day)



Daily Usage by Version (30 Day)



The PowerShell Ecosystem



The PowerShell Challenges



The PowerShell Challenges



How to start?

How to learn?

Ressources?



Which processes and
use cases should I
automate?

How to organize my
scripts?

How to standardize?

...



How to organize my
PowerShell scripts and
modules?

How to securely
manage credentials?

How to enable others
to consume my
scripts?

Does this remind you of the way PowerShell
is used in your organization today?



„De-centralized“ PowerShell landscape

PowerShell know-how bottleneck

No org-wide secure
credential management

PowerShell is only used
in expert circles

The potential of PowerShell
remains untapped



Unlock the Power
of PowerShell across your
organization



What it takes to successfully automate with PowerShell in the enterprise



Standardization

Centralization & Security

Easy accessibility for non-IT experts

Monitoring

Integration



Accessibility for non-IT experts



BEFORE

A screenshot of the Windows PowerShell ISE interface. The title bar says "Administrator: Windows PowerShell ISE". There are two tabs open: "Untitled3.ps1" and "Exchange_Webinar.ps1". The "Untitled3.ps1" tab contains a large block of PowerShell script code. The code is used to set up auto-replies for multiple mailboxes. It uses variables like \$cmdArgs, \$msg, and \$box, and performs operations like Set-MailboxAutoReplyConfiguration and Get-Mailbox. The code is color-coded for syntax highlighting. The status bar at the bottom shows "Completed" and "Ln 1 Col 36".

```
118 $cmdArgs.Add('ExternalAudience', $replyType)
119 $cmdArgs.Add('InternalMessage', $InternalText )
120 $cmdArgs.Add('ExternalMessage', $ExternalText )
121 if($PSCmdlet.ParameterSetName -eq "Schedule Auto Reply"){
122     $cmdArgs.Add('AutoReplyState', 'Scheduled')
123     $cmdArgs.Add('EndDate', $EndDate)
124     $cmdArgs.Add('StartTime', $StartDate)
125     $msg += "scheduled"
126 }
127 else{
128     $cmdArgs.Add('AutoReplyState', 'Enabled')
129     $msg += "enabled"
130 }
131
132 $Script:resHtml = @()
133 foreach($item in $MailboxIds){
134     try{
135         $box = Get-Mailbox -Identity $item -ErrorAction Stop
136         if($null -ne $box){
137             try{
138                 $null = Set-MailboxAutoReplyConfiguration @cmdArgs -Identity $box
139                 if($GenerateReport -eq $true){
140                     $Script:resHtml += Get-MailboxAutoReplyConfiguration -Identity $box
141                 }
142                 $resultMessage += [System.String]::Format($msg,$box.UserPrincipalName)
143             }
144             catch{
145                 Write-Output "Error occurred at set Mailbox $($item)"
146             }
147         }
148     }
149     catch{
150         Write-Output "Error occurred at get Mailbox $($item)"
151     }
152 }
```

AFTER

A screenshot of the ScriptRunner SelfService interface. The top navigation bar shows "1.company.net/OutOfOffice/" and "SelfService". The main page has a title "OutOfOffice" and a subtitle "Create OOF Notifications With ScriptRunner Integration". Below this, there are several input fields:

- "Please select the desired Parameter Set:" dropdown menu showing "Schedule Auto Reply".
- "Please select mailbox(es) Heiko was here:" search bar with results "Angus Young", "Shania Twain", and "Phil Collins".
- "Please select the notification template:" dropdown menu showing "Working From Home|Home Office".
- "Please select the start of absence" date input field showing "Feb 4, 2021, 5:21 PM".
- "Please select the end of absence" date input field showing "Feb 11, 2021, 5:21 PM".

A green play button icon is located in the bottom right corner of the form area.

Unlock the Power of PowerShell



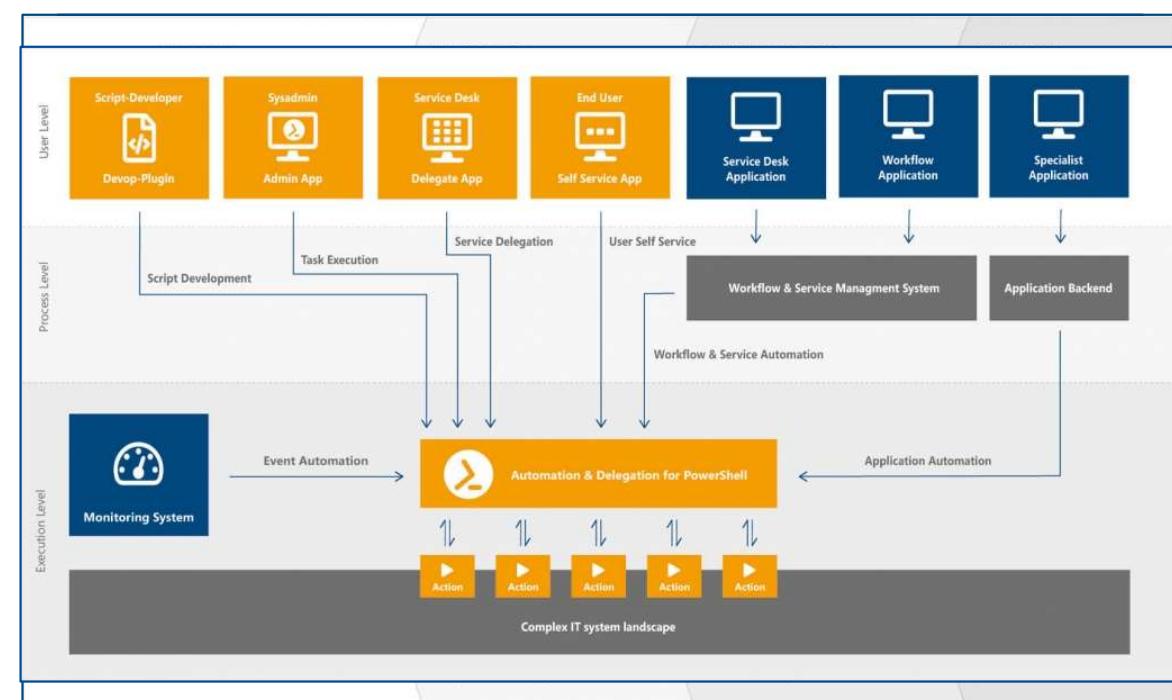
↗️ Centralize your PowerShell scripts and modules

🔒 Manage credentials & permissions securely

💻 Automatically transform PowerShell scripts into Web GUIs

👤 Delegate recurring tasks to help desk and end users

⚙️ Integrate with ITSM, Monitoring & Workflow systems and more



Hundreds of free
and ready-to-use
PowerShell scripts.

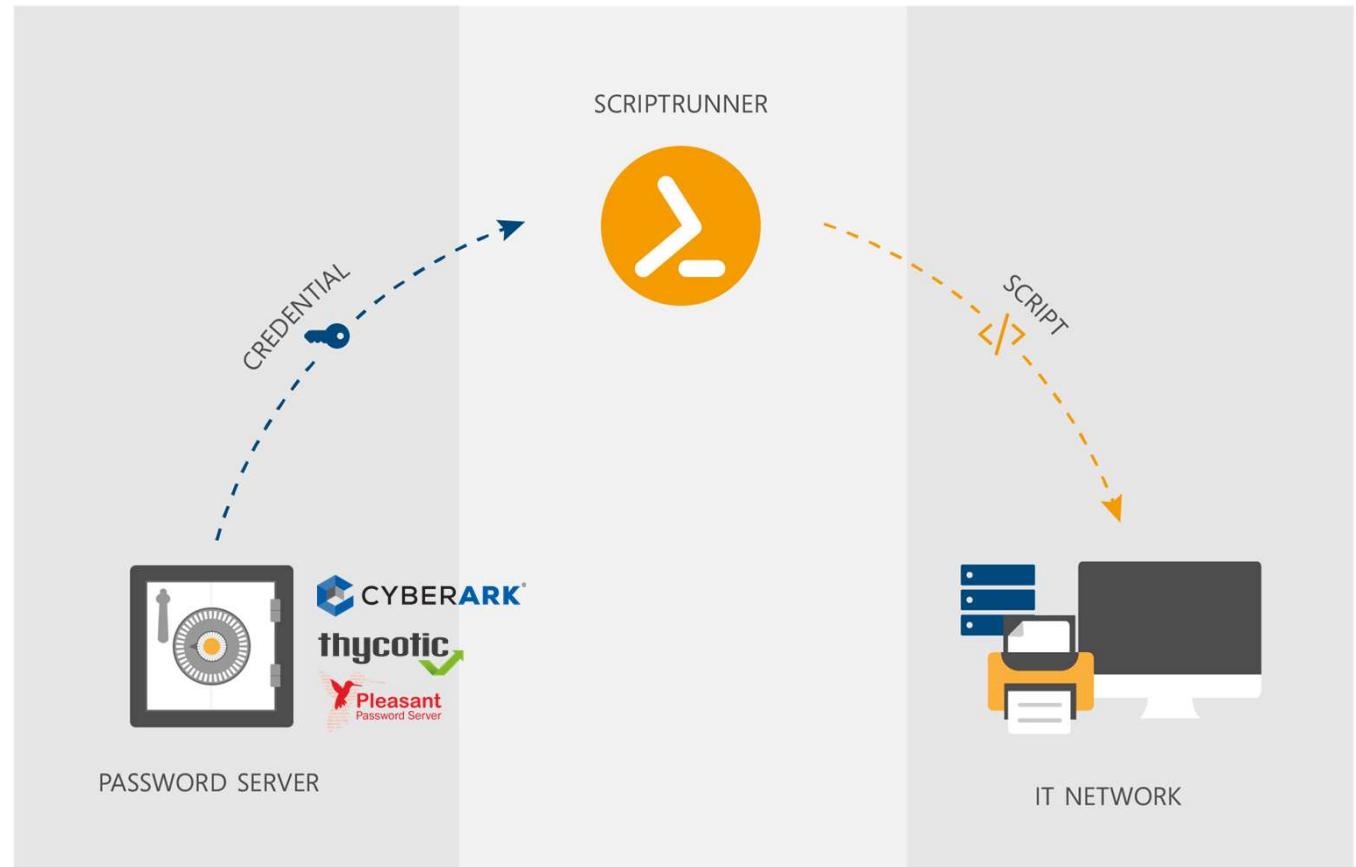
github.com/scriptrunner/ActionPacks



Password Server Support

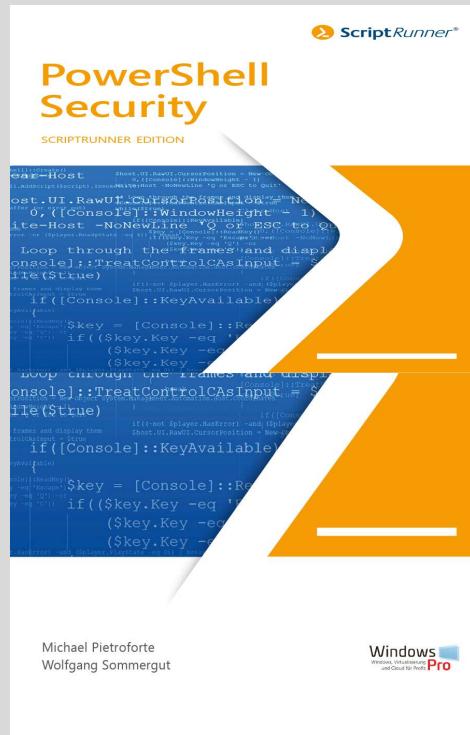


- Centralized password safes
 - CyberArk Password Vault
 - Pleasant Password Server
 - Thycotic Secret Server
- No credentials are stored locally
- Automatic password rotation
- One single secure password repository for multiple ScriptRunner instances



Free ebook: PowerShell Security

ip.scriptrunner.com/en/powershell-security-guide



PowerShell Security

Table of contents

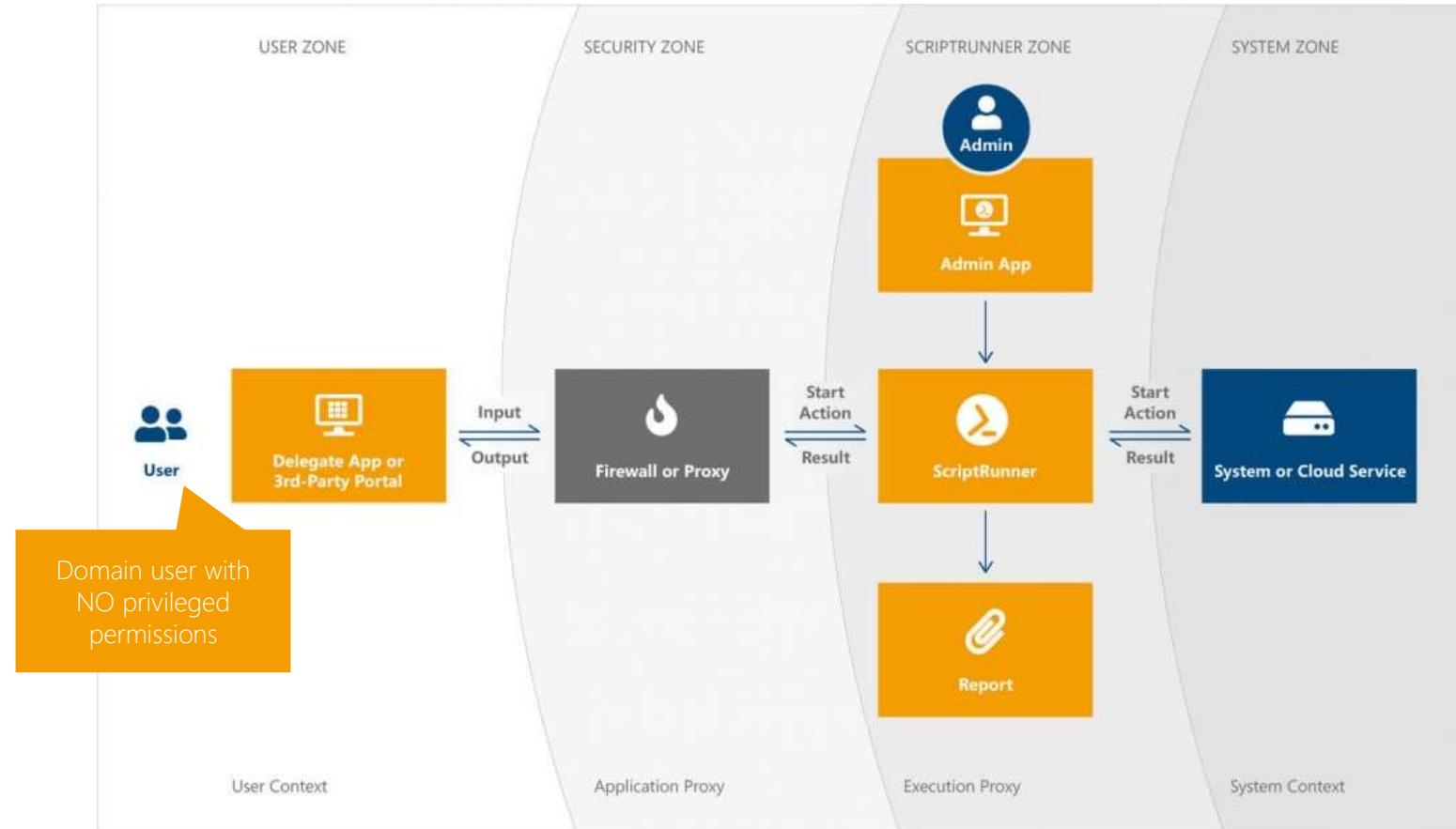
1	PowerShell as a hacking tool: Prevent abuse of scripts	8
1.1	Lax default configuration of PowerShell	9
1.2	Hacking tools for PowerShell	10
1.3	General blocking of PowerShell	12
1.4	Circumvention through alternative shells	14
1.5	Secure PowerShell with integrated mechanisms	15
2	Restrict execution of scripts	20
2.1	Setting an execution policy	20
2.2	Siging PowerShell scripts	25
2.3	Reduce PowerShell risks with Constrained Language Mode ..	36
3	Secure communication	48
3.1	Installing OpenSSH on Windows 10 and Server 2019	48
3.2	PowerShell remoting with SSH public key authentication	57
3.3	Creating a self-signed certificate	64
3.4	Remoting over HTTPS with a self-signed certificate	71
4	Just Enough Administration	81
4.1	JEA Session Configuration	81
4.2	Defining and assigning role functions	92
5	Audit PowerShell activities	98
5.1	Log commands in a transcription file	98
5.2	Scriptblock logging: Record commands in the event log	106

Demo

ScriptRunner



ScriptRunner – Secure Delegation



<https://www.scriptrunner.com/de/software/system/>

ScriptRunner test report fom a Microsoft MVP



Adam Bertram aka Adam The Automator
Microsoft MVP Cloud and Datacenter Management

"ScriptRunner is a master of organizing, categorizing and delegating scripts."

"ScriptRunner is a real DevOps tool for PowerShell scripters."

"Compared to large automation platforms, ScriptRunner is significantly cheaper"

"to get the same functionality of ScriptRunner, it would require several applications to get a similar feature set"

"If you are looking into automating your administration and support processes easily and securely, you will find a powerful and professional tool here."

<https://www.adamtheautomator.com/scriptrunner-bringing-powershell-to-devops/>

Unlock the Potential of PowerShell across your organization



Unlock the Power of PowerShell across your organization



Manager

Our organization becomes more productive

Deliver great scripts in a secure environment



DevOps

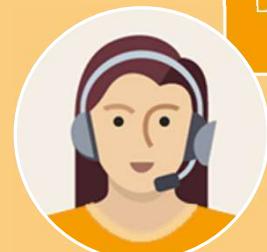
Delegate recurring tasks and save time



Administrator



Deliver more & better services in less time



HelpDesk

Develop scripts for more use cases/users



Developer



EndUser

Faster task completion with self-services

Put ScriptRunner
to the test.
30 days for free.

lp.scriptrunner.com/en/demo-download



Let's talk about your
automation use cases.
Book a online session
with me.

[lp.scriptrunner.com/meetings/heiko-
brenn/demo-en](https://lp.scriptrunner.com/meetings/heiko-brenn/demo-en)



Manage Microsoft Teams with PowerShell and save time.

Wednesday, June 23
16:00 CEST
10:00 pm EDT

[https://lp.scriptrunner.com/en/webinar-
microsoft-teams-2021](https://lp.scriptrunner.com/en/webinar-microsoft-teams-2021)



Put
ScriptRunner
to the test.
30 days for
free.

lp.scriptrunner.com/en/demo-download

www.scriptrunner.com



Ludwig-Erhard-Straße 2 | 76275 Ettlingen | Germany

Tel: +49 7243 20715-0
Mail: info@scriptrunner.com