



Manage Microsoft Teams with PowerShell & ScriptRunner

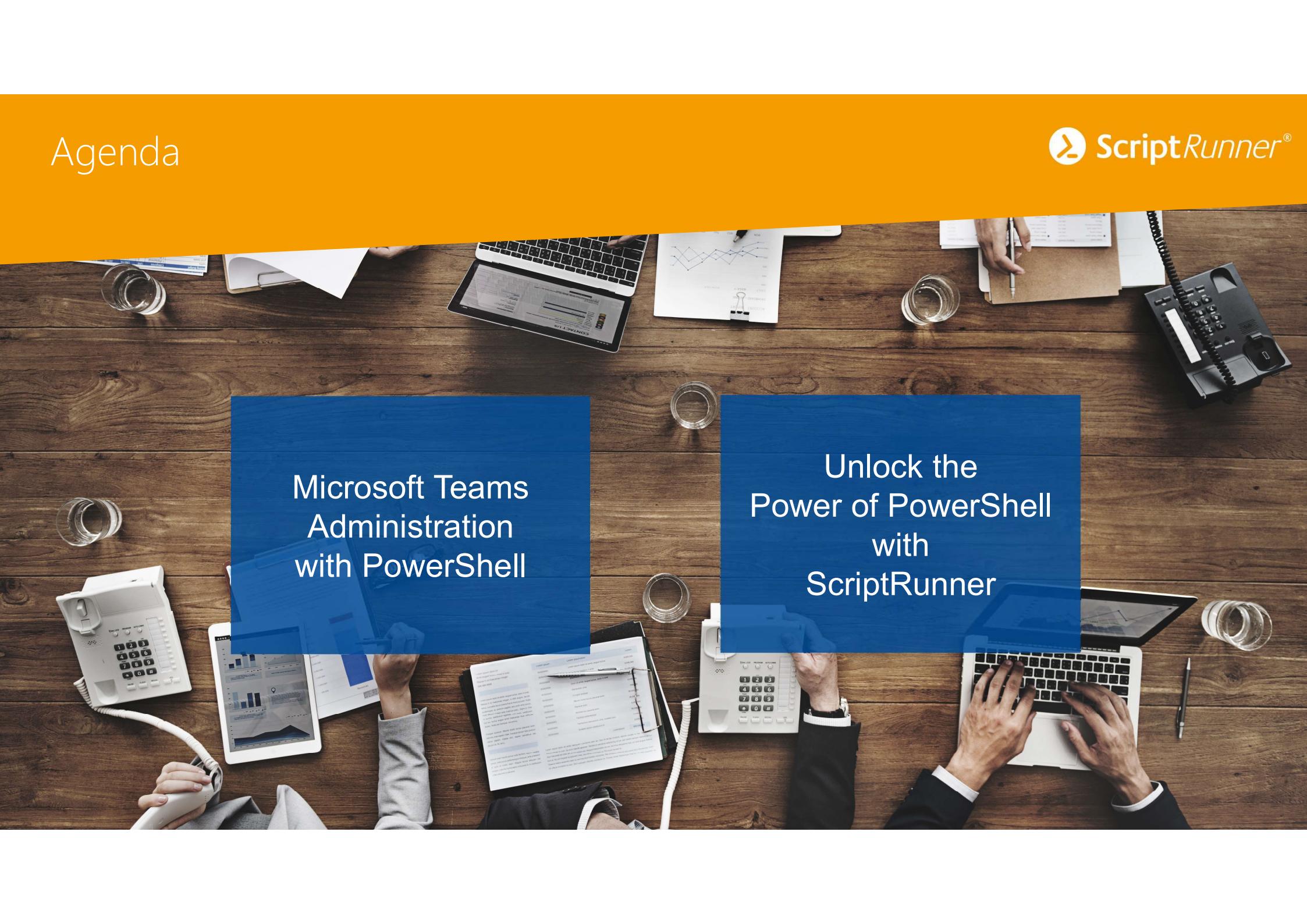
Automate. Delegate. Relax.

Heiko Brenn  @heikobrenn

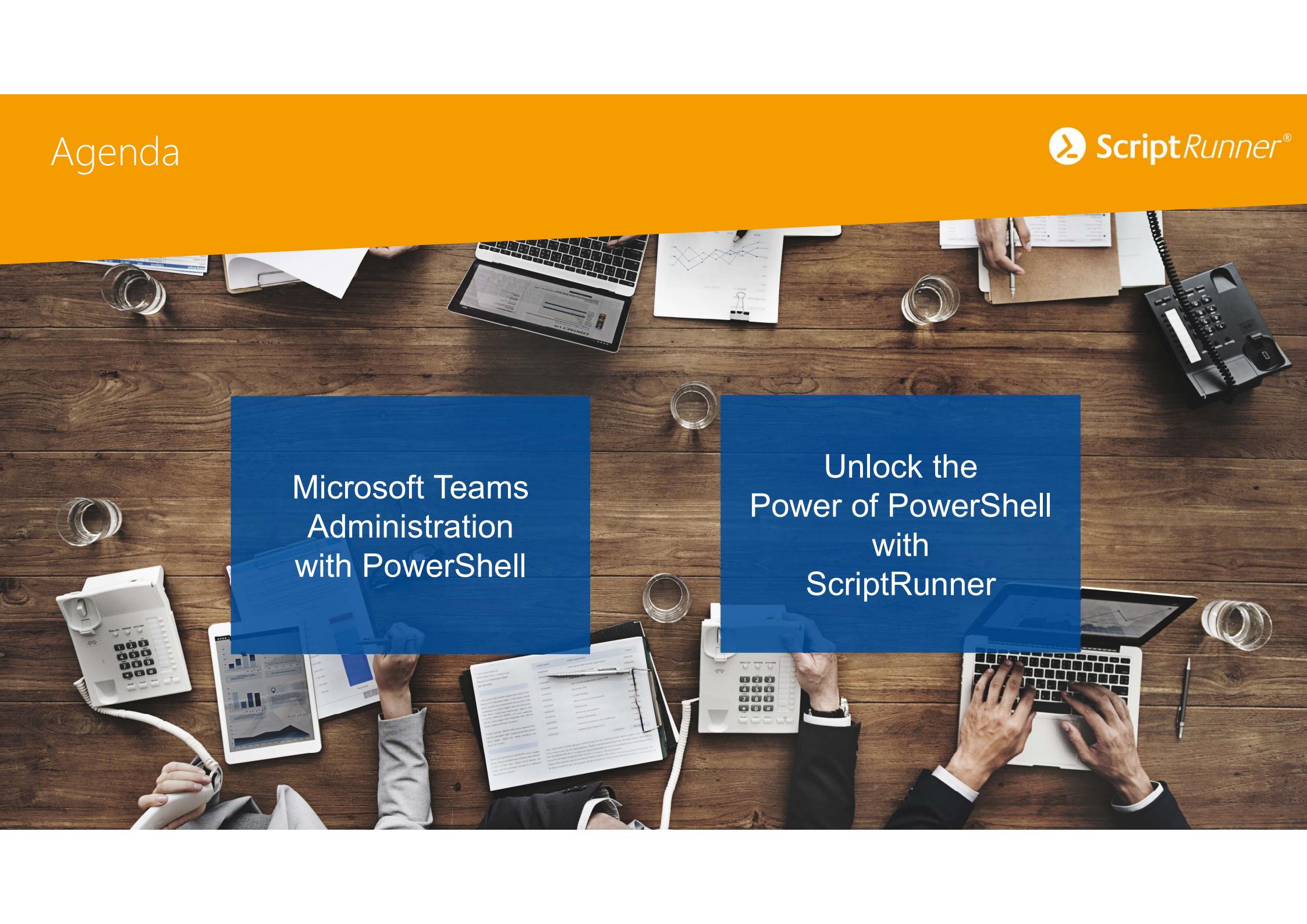
www.scriptrunner.com



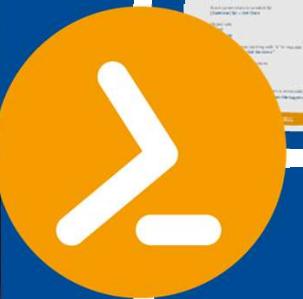
Agenda



Microsoft Teams
Administration
with PowerShell



Unlock the
Power of PowerShell
with
ScriptRunner



[linkedin.com/in/HeikoBrenn](https://www.linkedin.com/in/HeikoBrenn)

github.com/HeikoBrenn

+49-162-4879156

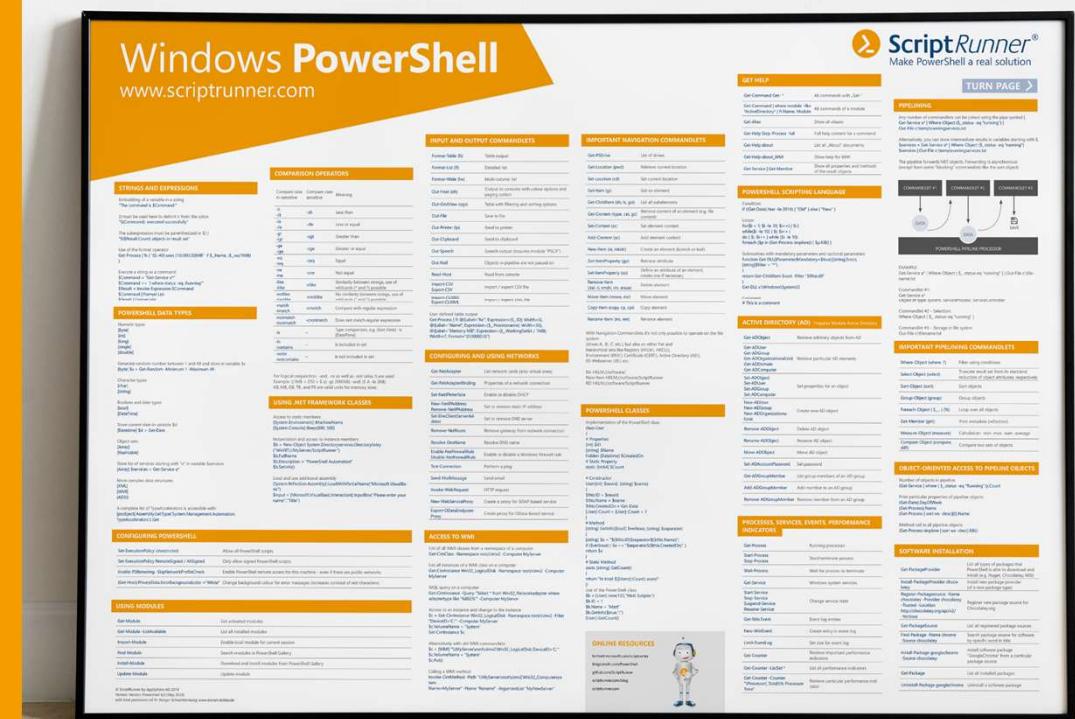
heiko.brenn@scriptrunner.com

ScriptRunner®

The #1 for PowerShell Management

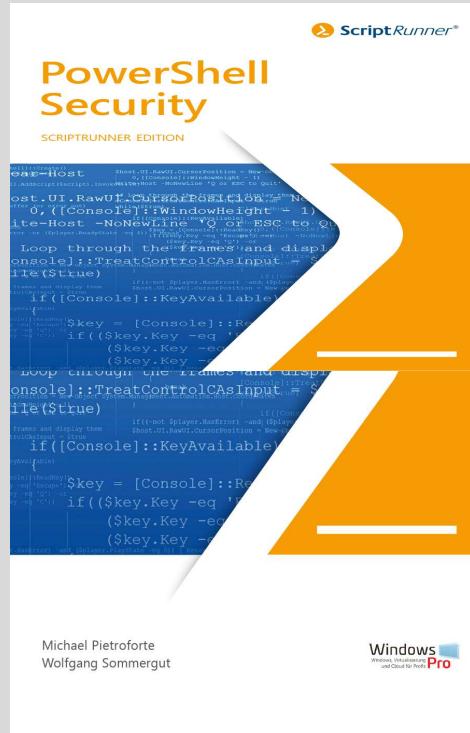
Get your free PowerShell poster

lp.scriptrunner.com/en/powershell-poster



Free ebook: PowerShell Security

lp.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	Signing 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

Teams PowerShell Module



Provisioning & management
of Teams/Channels/etc.

First version: 0.9.3 November 2017
Latest release: 2.0 March 2021

Number of cmdlets/functions: 568

ObjectIDs or GUIDs to identify teams
and Office 365 groups

Webhooks and PowerShell



Typical Use Cases



Create/Delete Teams/Channels

Add/Remove Users to/from
Teams/Channels

Manage Teams Settings

Get Current Teams/Channel Settings

Send Messages to Channels



Demo

Teams PowerShell Module



Cheat Sheet for Microsoft Teams PowerShell Module



Created by Damian Scoles

www.practicalpowershell.com

twitter.com/PPowerShell

PowerShell Quick Reference for Microsoft Teams [Ver. 1.0.5]		
Install Module (PS 5+) Install-Module MicrosoftTeams -Repository PSGallery	List All Teams Get-Team Find Single Team Get-Team where {\$_.DisplayName -eq 'IT Department'}	Complete Cmdlet List Add-TeamUser Connect-MicrosoftTeams Disconnect-MicrosoftTeams Get-CsPolicyPackage Get-CsUserPolicyPackage Get-CsUserPolicyPackageRecommendation Get-Team Get-TeamChannel Get-TeamHelp Get-TeamsApp Get-TeamUser Grant-CsUserPolicyPackage New-Team New-TeamChannel New-TeamsApp Remove-Team Remove-TeamChannel Remove-TeamsApp Remove-TeamUser Set-Team Set-TeamArchivedState Set-TeamChannel Set-TeamsApp
Load Teams PS Module Import-Module MicrosoftTeams -MinimumVersion 1.0.4	Create Team New-Team -MailNickname 'Marketing' -DisplayName 'Marketing' -Visibility 'Private' Remove a Team Remove-Team -GroupId '503b7c56-b15d-4b4c-8cca-09103984b2bb' Modify a Team Set-Team -GroupId '7a4fab2f-6a22-46d0-826b-a167fdc03892' -Visibility 'Public'	
Connect to Microsoft Teams with PowerShell Connect-MicrosoftTeams		
List all Commands for Teams Get-Command Where {\$_.Source -eq "MicrosoftTeams")		
Get-Help	Create New Microsoft Team	
Get-Help <command> Get-Help <command> -Examples Get-Help <command> -Full	New-Team -DisplayName "Marketing Campaign ~ Big Corp, Inc." New-Team -DisplayName "Project Rebuild" -Description "First project of 2020 (HR)" New-Team -DisplayName "New Test Group" -ShowInTeamsSearchAndSuggestions \$False	
Examples Get-Help New-Team Get-Help New-Team -Examples Get-Help New-Team -Full	Creating New TeamChannel (**) New-TeamChannel -GroupId 933fe926-555a-4832-87d1-8f700736e003 -DisplayName "Campaign Channel" New-TeamChannel -GroupId cfdba387-1319-4f6b-a883-8700046f07e7 -DisplayName "Internal Sales" ** GroupId is from the ID of an existing Team. To see the ID's, run Get-Team	
	PowerShell Gallery: https://www.powershellgallery.com/packages/MicrosoftTeams/ Documentation: https://docs.microsoft.com/en-us/powershell/module/teams/	
Configuring Team Settings		
Configuring Team Settings (notice the consolidation to the Set-Team cmdlet ... changed from 0.9.4 Beta) Set-Team -GroupId d36f235f-d30e-4460-98a2-5728b906fbfd -GiphyContentRating Strict -AllowStickersAndMemes \$False -AllowCustomMemes \$False Set-Team -GroupId cfdba387-1319-4f6b-a883-8700046f07e7 -AllowCreateUpdateChannels \$True Set-Team -GroupId cfdba387-1319-4f6b-a883-8700046f07e7 -AllowUserDeleteMessages \$False Set-Team -GroupId cfdba387-1319-4f6b-a883-8700046f07e7 -ShowInTeamsSearchAndSuggestions \$False Set-TeamChannel -GroupId cfdba387-1319-4f6b-a883-8700046f07e7 -CurrentDisplayName "Internal Sales" -Description "Sales Channel"		
Verify Settings	Microsoft Teams Cleanup	Cmdlet Notes
Get-Team Get-TeamChannel Get-TeamsApp Get-TeamUser	Remove-Team -GroupId '933fe926-555a-4832-87d1-8f700736e003' Remove-TeamChannel -GroupId cfdba387-1319-4f6b-a883-8700046f07e7 -DisplayName "Internal Sales" Remove-TeamUser -GroupId '933fe926-555a-4832-87d1-8f700736e003' -User <UPN>	Get-Team has no 'GroupId' parameter. In order to get an individual group, use something like this: <code>Get-Team / Where {\$_.GroupId -eq '933fe926-555a-4832-87d1-8f700736e003'}</code>
** Use ' ft' or ' fl' to format your results. **		

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



„Unorganized“ PowerShell usage

PowerShell know-how bottleneck

No org-wide secure
credential management

PowerShell is only used
in expert circles

The potential of PowerShell
remains untapped



What it takes to successfully automate with PowerShell in the enterprise



Standardization

Centralization

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 a web-based application titled "SelfService" under "Out of Office". The page has a header "1.company.net/OutOfOffice/" and a main title "OutOfOffice". Below the title is a sub-header "Create OOF Notifications With ScriptRunner Integration". The page contains several input fields and dropdown menus:

- "Please select the desired Parameter Set:" dropdown menu showing "Schedule Auto Reply".
- "Please select mailbox(es) Heiko was here:" search input field 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".

In the bottom right corner, there is a green circular button with a white play icon.

Unlock the Power of PowerShell



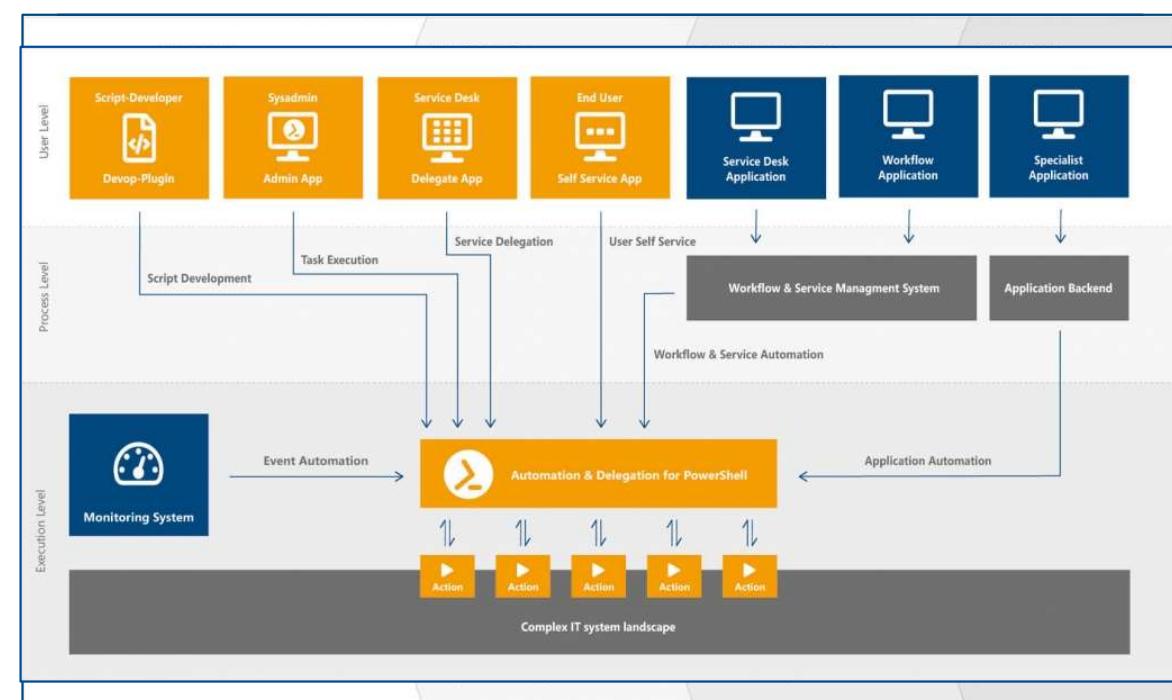
↗️ Centralize your PowerShell scripts and modules

🔒 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.

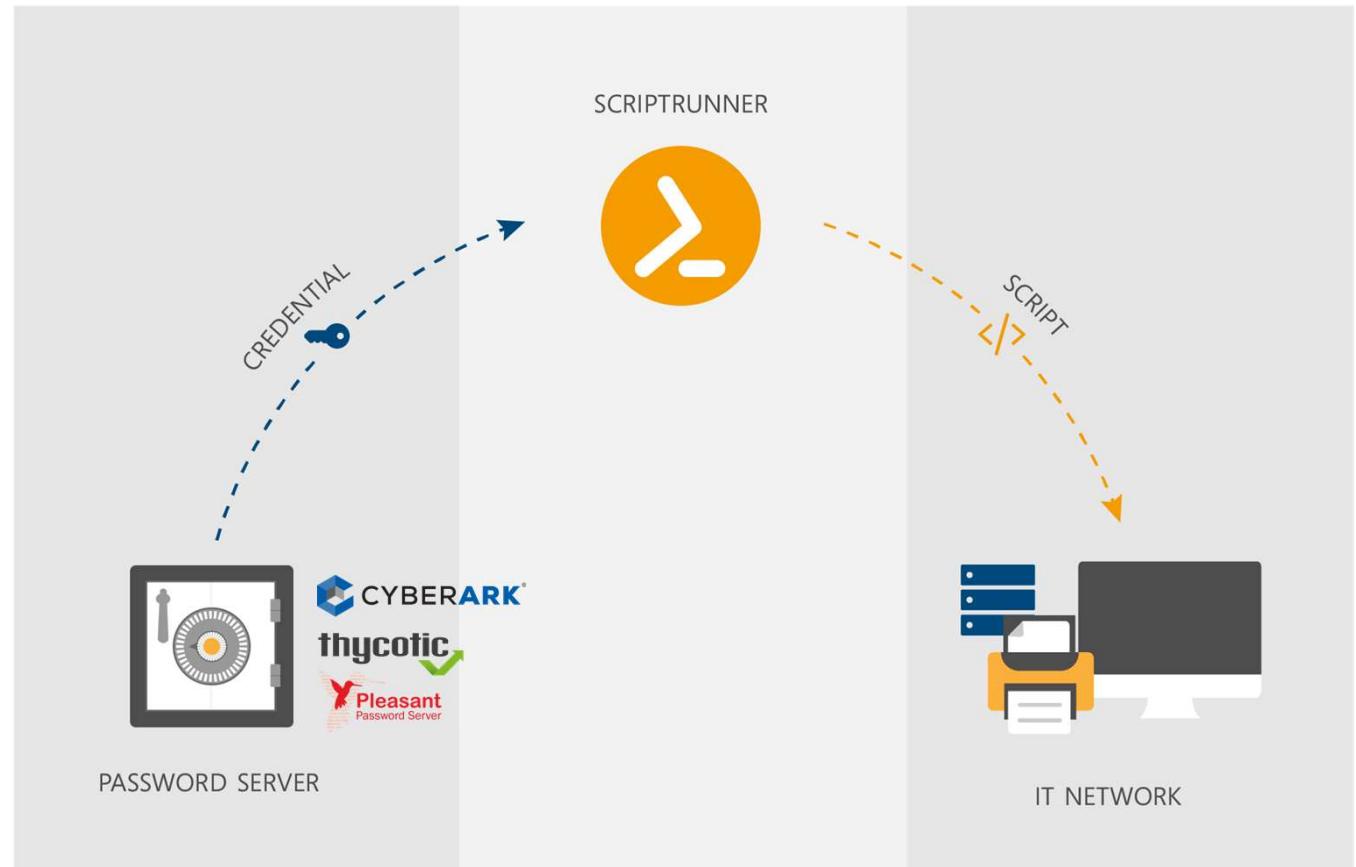
github.com/scriptrunner/ActionPacks



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



ScriptRunner – How to run scripts



Manually



Scheduled



Automatically

Web-based user interaction

Time-controlled tasks

Event triggered by 3rd party systems

Demo

ScriptRunner



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



Developer

Develop scripts for more use cases/users



HelpDesk

Deliver more & better services in less time



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)



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