PowerShell Conference Europe 2019

Hannover, Germany

June 4-7, 2019

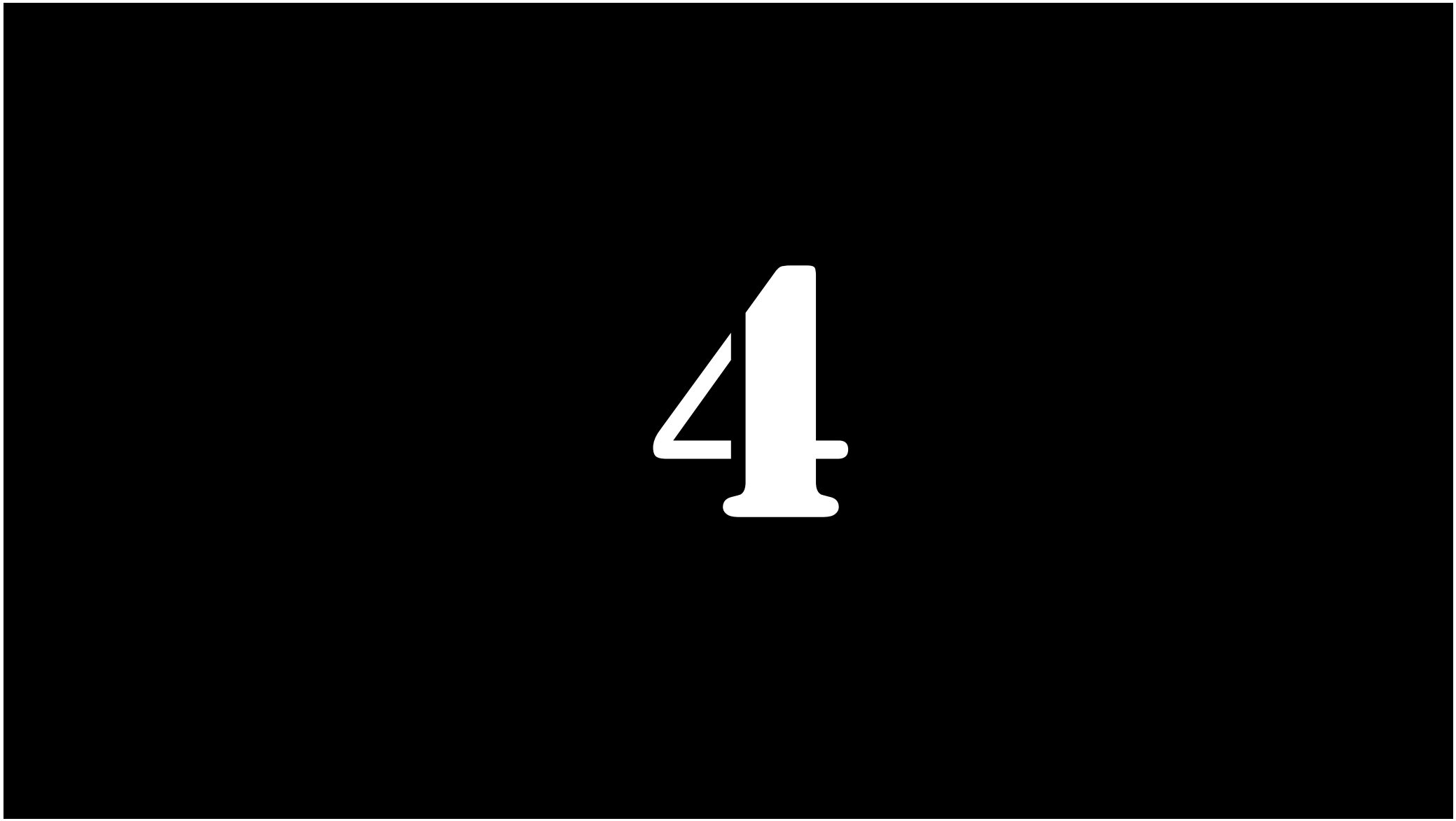
Posh-SSH and the Lessons Learned

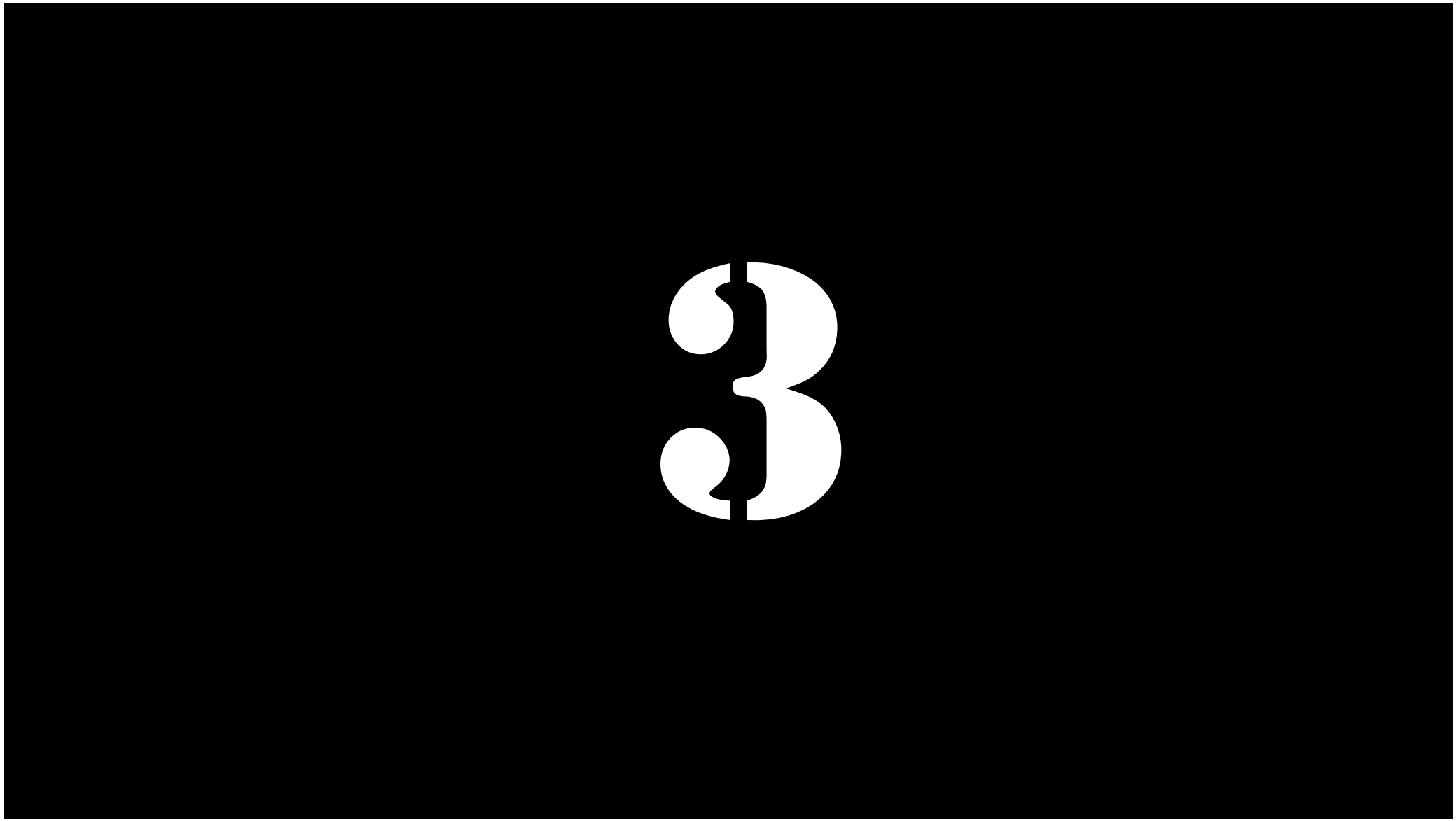
Carlos Perez

Platinum Sponsor

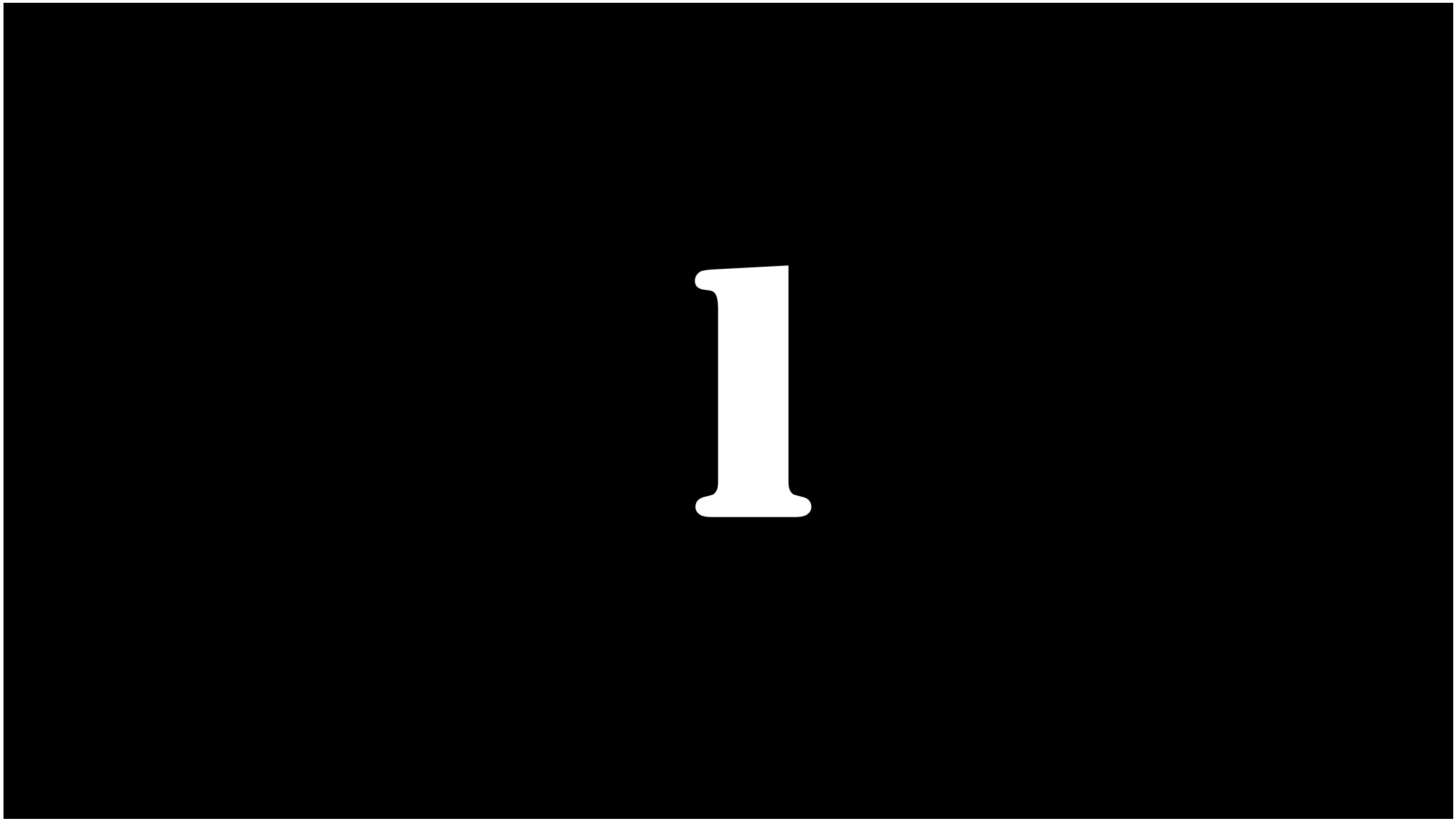
aws

Video operator, did you start the recording?


4


3

2



PowerShell Conference Europe 2019

Hannover, Germany

June 4-7, 2019

Posh-SSH and the Lessons Learned

Carlos Perez

Platinum Sponsor

This Session

- Why I wrote Posh-SSH and Design
- Challenges of a OpenSource Project
- Future of the project.

@Carlos_Perez

Why I wrote Posh-SSH

- Needed to automate tasks for vulnerability and zero day research on ESXi, CentOS and Ubuntu VMs.
- Wanted a project to learn C# ☺

Requirements

- It needed to allow me to automate a task on multiple server.
- Needed to support password and OpenSSH key authentication.
- Support to upload files:
  - SCP
  - SFTP
- Support for interactive shells.

Design

- As a security guy fingerprint verification was important.
- For Keyboard-Interactive authentication needed evening for which C# examples where available.
- Depend heavily on the SSH.Net Library https://github.com/sshnet/SSH.NET
- Module returns objects for most of its cmlets and functions.
- Supports multiple sessions to aid in automation of multiple systems.

DEMO

Posh-SSH

@Carlos_Perez

```powershell
#############################################################
# Create a connection to the servers using Username and Password #
#############################################################
$credentials = Get-credential
$ips = @("10.120.120.101", "10.120.120.102")
New-SSHSession -ComputerName $ips -credential $credentials
Get-Command -Noun sshtrustedhost
Get-SSHTrustedHost


#################################
# Authenticate with private key #
#################################
$key_credentials = Get-credential
$key = get-content -raw "C:\Users\carlos\.ssh\id_rsa"
New-SSHSession -ComputerName $ips -credential $credentials -KeyString $key
```

```powershell
##########################################
# Invoking single commands against binaries #
##########################################
$sessions = Get-SSHSession
Invoke-SSHCommand -Command "hostname" -SSHSession $sessions
Invoke-SSHCommand -Command "ls /tmp" -SessionId 0 | gm

# Show Inoke-SSHCommand does not maintain state like plink
Invoke-SSHCommand -Command "pwd" -SessionId 0
Invoke-SSHCommand -Command "cd /tmp" -SessionId 0
Invoke-SSHCommand -Command "pwd" -SessionId 0
```

```powershell
##################################################
# Working with a Shell like the ones in devices (Cisco, VyOS) #
##################################################
New-SSHSession 10.120.120.254 -Credential vyos
Invoke-SSHCommand -Command "show interfaces" -SessionId 4


Get-Command -Noun "sshstream*"


$shellStream = New-SSHShellStream -SessionId 4
Invoke-SSHStreamShellCommand -ShellStream $shellStream -Command "show
interfaces`n"


$shellStream | Get-Member
```

```
#######
# SCP #
#######
Set-SCPItem -ComputerName 10.120.120.101 -Credential carlos -Path .\sysmon.evtx -
Destination /tmp
Get-SCPFile -ComputerName 10.120.120.101 -Credential carlos -RemoteFile
/tmp/sysmon.evtx -LocalFile .\sysmon_new.evtx
```

```powershell
################
# SFTP Session #
################
New-SFTPSession -ComputerName $ips -credential $credentials -KeyString $key
Get-SFTPLocation -Session $SFTPSessions
Get-SFTPChildItem -SessionId 0
Get-SFTPChildItem -SessionId 0 | Get-Member
Set-SFTPLocation -SessionId 0 -Path "/tmp"
Get-SFTPLocation -SessionId 0
```

```
 ########################
# SFTP Upload/Download #
########################
cd C:\Users\carlos\Desktop
Set-SFTPItem -SessionId 0 -Destination /tmp -Path C:\Users\carlos\Desktop\backup.7z
Get-SFTPChildItem -SessionId 0 -Path /tmp | Where-Object FullName -Like *.7z
```

```
 ############################
# SFTP working with files #
############################


Get-SFTPChildItem -SessionId 0 -Path /tmp | Where-Object FullName -Like *.7z
Get-SFTPPathAttribute -SessionId 0 -Path "/tmp"
Get-SFTPPathAttribute -SessionId 0 -Path "/tmp/backup.7z"
Remove-SFTPItem -SessionId 0 -Path "/tmp/backup.7z"
```

Mistakes – Project Management

- Need better documentation with more examples.
- Should have done template for GitHub Issues.
- Should have done automated testing early on.
- Never make a promise in GitHub issues.

Mistakes – Code

- Better support for pipeline
- Release major versions more often so as to push improvement that break backward compatibility.
- More unit tests.
- Documentation on how to submit PRs.
- Automate module build and release.

Summary

- If you publish to the PowerShell gallery people expect you to maintain it. Publishing and forgetting hurts the ecosystem.
- Set expectations early on release tempo and breaking features.
- Unit testing/Automated testing sucks but it is a must.
- Good initial practices go a long way to mitigate maintenance pain down the road.

Slides and demo code

```
Start-Process -FilePath https://github.com/psconfeu/2019
```

@Carlos_Perez

Questions?

Use the conference app to vote for this session:
https://my.eventraft.com/psconfeu

@Carlos_Perez

about_Speaker

Research Lead at Trustedsec

Microsoft MVP

Active in the Security community.

@Carlos_Perez