

Erfassung von AutoPilot Diagnosedaten

Um verlässliche Daten und Logs eines AutoPilot und/oder PPD (aka WhiteGlove) Enrollments zu bekommen, habe ich ein paar einfache Skripte geschrieben.

Ein Satz dieser Skripte dient für die LOKALE Erfassung der Daten auf einer frisch ausgerollten Box, egal ob erfolgreich (grüner Schirm) oder fehlerhaft (roter Schirm).

Ein weiteres benötigt lesenden Zugang zum Tenant (MEM, Intune) und liefert zwei sehr aussagefähige CSV Dateien zu den Geräten und zu den Ereignissen im Intune, die sich prima mit Excel auseinandernehmen lassen.

Die Skripte basieren auf der „eingebauten“ *mdmdiagnosticstool.exe* und dem *WindowsAutoPilotIntune* PowerShell Modul von Michael Niehaus sowie dem *Microsoft.Graph.Intune* PowerShell Modul.

„Schmankerl“:

Wenn ein USB-Stick angeschlossen ist (Empfehlung) und man die Teile von dort startet, schreiben die Teile ihre Daten in ein Verzeichnis auf dem Stick (mit der Seriennummer der Box im Namen, auf der sie ausgeführt werden).

Ansonsten schreiben sie ins %temp% des jeweiligen Benutzers, der sie ausführt.

Lokale Erfassung



Empfehlung: unbedingt den QR-Code erfassen (dafür gibt es ja zahlreiche Apps 😊)

„Shift“ F10 drücken, dann sollte ein CMD Prompt starten.

Darin dann (z.B. vom USB Stick aus) das Skript *get-local-autopilot-infos.cmd* ausführen, das ruft intern die o.g. *MdmDiagnosticsTool.exe* auf und startet danach ein weiteres PowerShell Skript (*Get-AutoPilotDiagnostics* von Michael Niehaus).

Letzteres kann man wie folgt installieren oder von einer Box kopieren, wo es bereits installiert war (das habe ich hier gemacht & selbiges in den gleichen Ordner gepackt, wo das aufrufende CMD liegt 😊)

```
Install-Script -Name Get-AutopilotDiagnostics
```

Quelle:

[PowerShell Gallery | Get-AutopilotDiagnostics 5.6](#)

Das Ganze funktioniert natürlich auch OHNE PPD / WhiteGlove Ablauf 😊

PowerShell Modul Microsoft.Graph.Intune

```
Install-Module Microsoft.Graph.Intune -Force
```

Quelle:

[PowerShell Gallery | Microsoft.Graph.Intune 6.1907.1.0](#)

Auf einer bereits komplett durchgelaufenen Box benötigt man allerdings Admin-Rechte für die Ausführung der Teile, was ja in der OOB Phase (s.o.) gegeben ist 😊

Noch'n Schmankerl:

Das *get-local-autopilot-infos.ps1* macht auch folgendes:

```
# copy json files
```

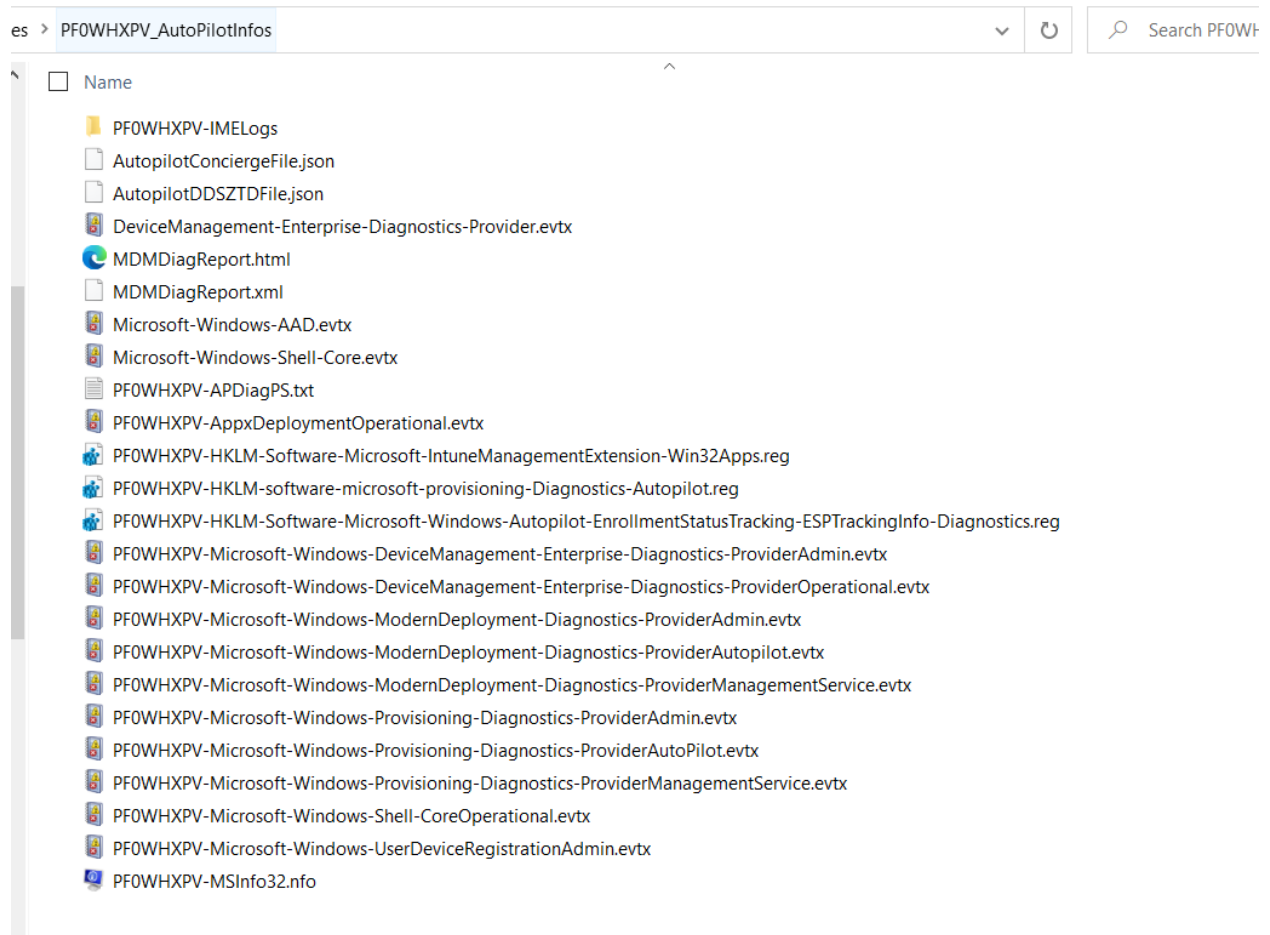
```
$CMD1 = "copy C:\Windows\ServiceState\wmansvc\*. * " + $OutAPDir
```

Es kopiert also die während des „Enrollments“ entstandenen JSON files ebenfalls in das Output Dir, war nen echter „Geheimtip“ von Mike, auch diese Dateien sind ziemlich aussagefähig 😊

Des Weiteren werden einige relevante EVTX File generiert, die das mdmdiagnostics.exe Tool nicht oder nur instabil (😞) erfasst.

Und auch alle IME (Intune Management Engine) Logs werden kopiert 😊

Am Schluss werden dann auch noch einige relevante Registry-Keys (im .reg Format) exportiert 😊



Typischer Fileset nach Ausführung des Skripts

Und noch'n Schmankerl:

Wenn man / frau die ganzen XMLs und JSONs wirklich lesbar haben möchte & auch auseinandernehmen und ggf. editieren möchte oder muss, nehme man / frau folgendes Tool, was ich „wiederentdeckt“ habe:

Den [JSON Buddy](#) 😊

Das Ding kann u.a. auch XML 2 JSON Migration vice versa 😊

Und es hat eine EinsA GUI 😊

Und ist in der Pro Version durchaus bezahlbar 😊

Erfassung der Tenant-Daten (Online)

Hierzu benötigt man einen Zugang zum jeweiligen Tenant und mindestens das PowerShell Module *WindowsAutoPilotIntune* (von Mike)

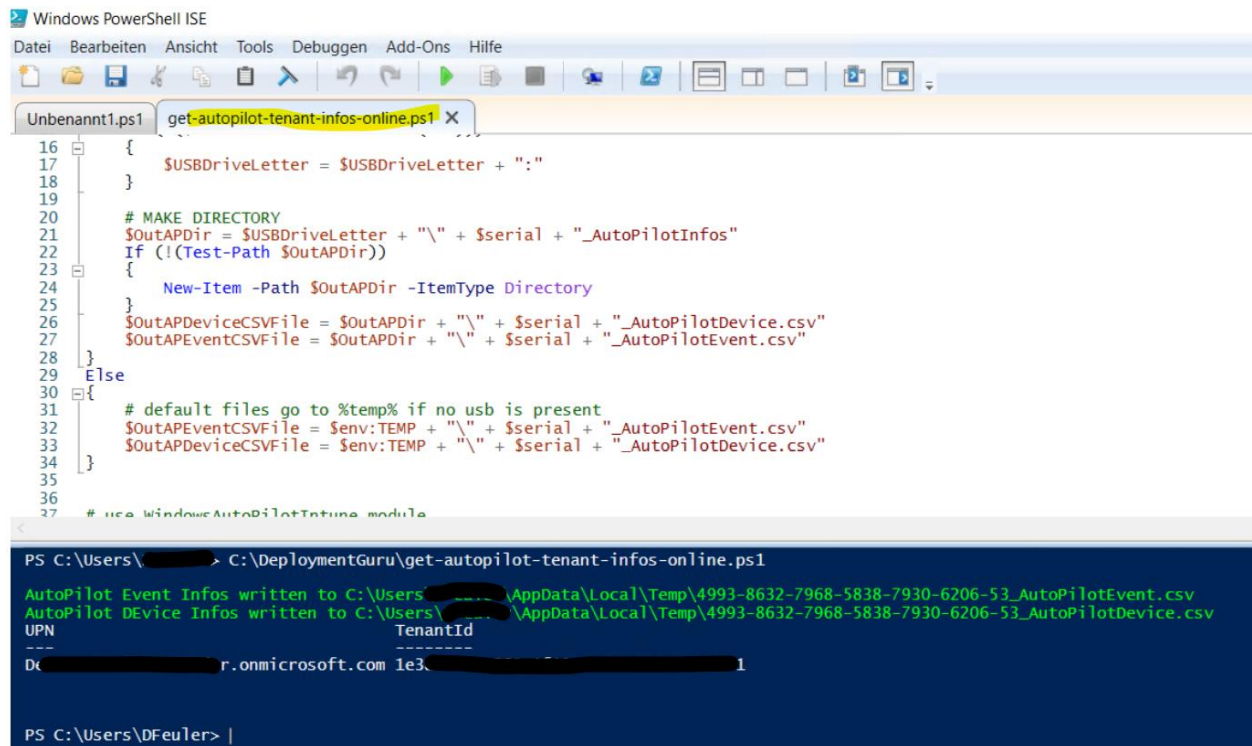
Selbiges kann man wie folgt installieren:

Install-Module -Name WindowsAutoPilotIntune

Quelle:

[PowerShell Gallery | WindowsAutoPilotIntune 4.8](#)

Das Teil gibt am Ende aus, wo es die CSVs abgelegt hat, wie gesagt bevorzugt erneut auf einen USB Stick, ansonsten halt ins %temp% des aufrufenden Benutzers, z.B.:



```
Windows PowerShell ISE
Datei Bearbeiten Ansicht Tools Debuggen Add-Ons Hilfe

Unbenannt1.ps1 get-autopilot-tenant-infos-online.ps1 X
16 {
17     $USBDriveLetter = $USBDriveLetter + ":"
18 }
19
20 # MAKE DIRECTORY
21 $OutAPDir = $USBDriveLetter + "\" + $serial + "_AutoPilotInfos"
22 If (!(Test-Path $OutAPDir))
23 {
24     New-Item -Path $OutAPDir -ItemType Directory
25 }
26 $OutAPDeviceCSVFile = $OutAPDir + "\" + $serial + "_AutoPilotDevice.csv"
27 $OutAPEventCSVFile = $OutAPDir + "\" + $serial + "_AutoPilotEvent.csv"
28 }
29 Else
30 {
31     # default files go to %temp% if no usb is present
32     $OutAPEventCSVFile = $env:TEMP + "\" + $serial + "_AutoPilotEvent.csv"
33     $OutAPDeviceCSVFile = $env:TEMP + "\" + $serial + "_AutoPilotDevice.csv"
34 }
35 }
36
37 # use WindowsAutoPilotIntune module

PS C:\Users\...> C:\DeploymentGuru\get-autopilot-tenant-infos-online.ps1
AutoPilot Event Infos written to C:\Users\...\AppData\Local\Temp\4993-8632-7968-5838-7930-6206-53_AutoPilotEvent.csv
AutoPilot Device Infos written to C:\Users\...\AppData\Local\Temp\4993-8632-7968-5838-7930-6206-53_AutoPilotDevice.csv
UPN                               TenantId
---                               -
Dc...r.onmicrosoft.com 1e3...1

PS C:\Users\DFeuler>
```

Hilfreiche Links zur Analyse der Daten

Wie man die resultierenden Daten „auseinandernimmt“, haben Mike & Anoop ja schon sehr ausführlich in ihren Blogs beschrieben 😊

Empfehlung hier:

das EVTX vom „Shell-Core“ ist COOL 😊

[Windows Autopilot diagnostics: Digging deeper – Out of Office Hours \(oofhours.com\)](#)

[Troubleshooting Windows device enrollment problems in Microsoft Intune - Intune | Microsoft Docs](#)

[MDM Diagnostics Tool - Autopilot Troubleshooting 5 Easy Tips \(anoopcnaair.com\)](#)