

Script "parse-ActiveSyncDebugLog" 2021-10-19 Exchange Server 2010 bis 2019

<u>Inhalt</u>

Die Idee	2
Das Problem	2
Die Microsoft-Lösung	2
Probleme mit der Microsoft-Lösung	4
Meine Lösung: "parse-ActiveSyncDebugLog"	4
Anwendung des Scriptes	4
Device-Suche und Object-Ausgabe	4
Gridview-Ausgabe	5
(formatierte) Tabellen-Ausgabe	6
HTML-Ausgabe	6
weitere Parameter und deren Kombinationen	6



Die Idee

Das Problem

Exchange Server speichern Mailboxen, auf die wir gerne mit unseren Mobile Devices zugreifen. Mit einer richtigen Konfiguration ist das auch einfach möglich. Aber manchmal verweigern Smartphones die Synchronisierung. Für eine Problemdiagnose liegen dann oftmals keine sinnvollen Fehlermeldungen oder Logfiles im Gerät vor. Hier kann der Exchange Server helfen.

Die Microsoft-Lösung

Im on-premise Exchange Server kann ab der Version 2010 das so genannte ActiveSyncDebugLog aktiviert werden. Dieses stellt dann sehr detailliert die Kommunikation zwischen Endgerät und dem ClientAccessService dar.

Wichtig sind an dieser Stelle folgende Informationen:

- Das Logging ist nicht standardmäßig aktiv
- Das Logging muss pro Mailbox aktiviert werden.
- Ein aktiviertes Logging wird sehr viele Logdaten pro Mailbox sammeln.
- Daher wird ein aktiviertes Logging vom Exchange Server nach 72 Stunden automatisch deaktiviert.
- Das Logfile liegt als Fließtext vor und kann über die PowerShell abgefragt werden.

So schaut es im Normalfall aus. Das Logging ist deaktiviert:

ActiveSyncAllowedDeviceIDs		
ActiveSyncBlockedDeviceIDs		
ActiveSyncMailboxPolicy	: Default 🥢	
ActiveSvncMailboxPolicvIsDefaulte	: True	
ctiveSvncDebuaLogaina	: False	
ctiveSyncEnabled	: True	
as ActiveSyncDevicePartnership	True	
asaccivesyncoeviceral chership	. 11 ue	

Mit der PowerShell kann das Logging aktiviert werden. Der Zustand wird in meinem Exchange Server aber nicht korrekt dargestellt:

PS C:\>> Set-CASMailbox -Identity '	'tessa.test" -ActiveSyncDebugLogging \$true
P5 C:\>> Get-CASMailbox -Identity '	'tessa.test" Format-List -Property "*activesync*"
ActiveSyncAllowedDeviceIDS ActiveSyncBlockedDeviceIDS ActiveSyncBlockedDeviceIDS ActiveSyncMailboxPolicyISDefaulted ActiveSyncEnabled ActiveSyncEnabled HaSActiveSyncEvicePartnership ActiveSyncSuppressReadReceipt	: {} : {} : Default : True : Fralse : True : True : True : False

Nach der Aktivierung kann dann das Problemgerät erneut synchronisiert werden. Das Ergebnis kann mit der PowerShell abgefragt werden. Bei der Abfrage wird der Schalter -GetMailboxLog benötigt. Damit wird die zusätzliche EigenschaftMailboxLogReport ausgelesen:

PS C:\>> Get-MODITeDeviceStatistic	s -Malibox tessa.test -GermaliboxLog
RunspaceId FirstSyncTime LastPolicyUpdateTime LastSyncAttemptTime LastSyncAttemptTime LastSuccesSSync DeviceType DeviceUpeSentTime DeviceWipeRequestTime DeviceWipeRequestTime DeviceWipeRequestTime AccountonlyDeviceWipeRequestTime AccountonlyDeviceWipeRequestTime AccountonlyDeviceWipeRequestTime AccountonlyDeviceWipeRequestTime AccountonlyDeviceWipeRequestTime AccountonlyDeviceWipeRequestTime AccountonlyDeviceWipeRequestTime AccountonlyDeviceWipeRequestTime AccountonlyDeviceWipeRequestTime DeviceWipeRequest LastPingHeartbeat RecoveryPassword DeviceTmel DeviceFoneNumber DeviceOS DeviceOS DeviceOSnequage DeviceOS	<pre>b46732b3-9ad1-47f5-9060-06c017fbb486 30.09.2020 07:24:57 30.09.2020 07:24:59 19.10.2021 16:18:10 19.10.2021 16:18:10 WindowsAall 1551172317744E297C336Fc48425F3A MSFT-WIN-3/10.0.18363 ******** Virtual Machine M-CL1 Windows 10.0.18363 German</pre>
	Log Entry: 0
	Danuastrima ·
	10/19/2021 18:01:26
	ServerName :

Das MailboxLog ist ein Fließtext. Hier habe ich den Text mal zur Analyse extrahiert. Die einzelnen Verbindungen zwischen dem Device und dem Exchange Server sind als Log Entry aufgezeichnet. In jedem Log Entry gibt es dann mehrere Attribute mit Werten. Die Attribute habe ich mal farblich markiert:



Durch diese Darstellung wird deutlich, wie ActiveSync funktioniert:

- Der Clients schickt einen Request (RequestBody) zum Server. Darin ist auch eine Authentifizierung (im RequestHeader) enthalten.
- Der Server prüft die Authentifizierung (AccessState, AccessStateReason) und senden eine Antwort (ResponseHeader, ResponseBody) zum Client.

WS IT-Solutions Script "parse-ActiveSyncDebugLog" 2021-10-19 Exchange Server 2010 bis 2019

Probleme mit der Microsoft-Lösung

Die Responses können sehr groß werden. Auf meinem Testclient wurde von einem vollständig im Sync stehenden Device innerhalb von 15 Minuten 2300 Zeilen Log generiert. Ebenso ist das Auffinden von Fehlern in dieser Darstellung nicht wirklich einfach.

Dazu kann ein Mailboxbenutzer durchaus mehrere ActiveSync-Devices mit seiner Mailbox verbunden haben. Die Logfiles werden pro Device gespeichert. Ohne Angabe des Devices werden also die Logdaten mehrerer Devices miteinander vermischt. Vorab muss also das ggf. passende Device gesucht werden.

Meine Lösung: "parse-ActiveSyncDebugLog"

Die Suche nach dem richtigen ActiveSync-Device ist kein wirkliches Problem. Das Lesen und Analysieren des Logtextes aber schon. Und da setzt meine Lösung auf.

Der Text im Logfile folgt einer gewissen Struktur. Meine PowerShell-Funktion parst diese Struktur und erstellt daraus ein PowerShell-Object. Dieses lässt sich wie gewohnt filtern, sortieren bzw. weiterverarbeiten. Dazu habe ich verschiedene Ausgabeoptionen, wie eine optimierte Tabellenausgabe und eine GridView-Ausgabe integriert. Und eine Suche für das ActiveSync-Device ist natürlich auch dabei 🙄.

Wie gewohnt sind im oberen Teil des Scriptes die Arbeitsfunktionen enthalten. Meine Funktion "verbinde-MX" stellt die Verbindung zum Exchange Server her. Im unteren Teil habe ich einige Beispielaufrufe abgelegt. Und natürlich ist auch die Aktivierung des Loggings mit dabei:



Anwendung des Scriptes

Device-Suche und Object-Ausgabe

Nachdem das Logging aktiviert und ein Sync durchgeführt wurde, kann die Funktion für die Mailboxbenutzerin in meinem Beispiel aufgerufen werden. Wie eingangs erwähnt kann ein Benutzer mehrere Devices verbunden haben. Und hier gibt es dann verschiedene Szenarien:

- Findet meine Funktion kein Device, wird einfach eine Fehlermeldung zurückgegeben.
- Wird nur ein Device gefunden, dann wird für dieses Device direkt die Ausgabe des Logfiles angezeigt.
- Sind mehrere Devices registriert, dann wird über ein GridView die Device-Abfrage gestartet. Es sind hier ein oder mehrere Devices möglich. Die Informationen zum Auffinden des richtigen Devices werden natürlich mit angezeigt:



Script "parse-ActiveSyncDebugLog" 2021-10-19 Exchange Server 2010 bis 2019

PS C:\>> p	arse-ActiveSy	ncDebugLog -Mailbox "tessa.te	est"					
	🛃 Wähle ein oo	der mehrere ActiveSync-Devices aus:					- 0	×
	Filter							\sim
	+ Add criteria 🔻	•						
	FriendlyName	DeviceId	Devicelmei	DeviceOS	DeviceUserAgent	DeviceType		
	M-CL1	155117C2317744E297C336FC48425F3A		Windows 10.0.18363	MSFT-WIN-3/10.0.18363	WindowsMail		
	M-PROD-CL1	A4C2BDCC8349BCF0C886EC720F29FF0C		Windows 10.0.18363	MSFT-WIN-3/10.0.18363	WindowsMail		
							OK	Cancel

Danach wird das geparste Logfile im Listenformat ausgegeben. Man erkennt sehr schön die einzelnen Bestandteile aus dem Logfile. Links sind die Attribute und rechts deren Werte notiert. Je Log-Entry wird ein Object ausgegeben:

DeviceID	: 155117C2317744E297C336FC48425F3A
DeviceFriendlyName	: M-CL1
DeviceModel1	
Device05	: windows 10.0.18363
Status	: DeviceOk
LogEntry	: 18
RequestTime	: 2021-10-19 18:18:10
ServerName	: M-MX2
AssemblyVersion	: 15.01.1913.003
Identifier	: 3FECA68E
URT	: POST //wicrosoft-Server-ActiveSvnc/Proxv/default.eas?cmd=Svnc&DeviceId=155117C2317744E297C336FC48425E3A&DeviceTvpe=WindowsMail
Cache-Control	; no-cache
Pragma	: no-cache
Content-Length	2.0
Accept-Encoding	; gzip
Accept-Language	
Authorization	*****
Host	: m-mx2.crashwork.global:444
X-FF-ClientIP	192.168.101.254
X-Forwarded-For	192.168.101.254
X-Forwarded-Port	: 13461
X-MS-EdgeIP	
X-ExCompId	: ClientAccessFrontEnd
MS-ASProtocolversion	: 16.1
X-M5-PolicvKev	: 1863221472
x-OriginalRequestHost	: mail.crashwork.de
X-OriginalRequestHostSchemePort	: 443:https:mail.crashwork.de
X-M5ExchangeActivityCtx	: v=1.0.0.0:Id=54b8921d-f37e-4184-96cb-b7734acb1e90:C=:P=
X-FAS-Proxy	: 5-1-5-21-2335075080-885438205-1618318378-1117.crashwork\tessa.test
msExchProxvUri	: https://mail.crashwork.de/Microsoft-Server-ActiveSync?oOAHBBAYURfCMXdE4pfDNvxI0]86B0CADm8LV2]uZG93c01haWw=
X-IsFromCafe	:1
X-SourceCafeServer	: M-MX2.CRASHWORK.GLOBAL
X-CommonAccessToken	: VQEAVAdXawSkb3dzowBBBUJhc2liTBRicmEzaHdvcmtcdGVzc2EudGVzdEutUv0xLTUTMiEtMiMzNTA3NTA4MC04ODU0MzqvMDUtMTYxODMX0DM30C0xMTE3Rw0A4
	WŻUWODAŁODGINDMAM JALI TEZMTÓŻNITÓZNI ZGUNTEZBWAAAAADTI. TELMSÓWBWAAAAADTI. TELNSÓWBWAAAAADTI. TELNSÓWBWAAAADTI. TELNSÓWBWAAADTI. TELNSÓWBWAAAADTI. TELNSÓ

Die Daten können z.B. mit Where-Object oder Export-CSV über die Pipeline einfach weiterverarbeitet werden.

Gridview-Ausgabe

Für einen schnellen Blick ist das PowerShell.Object eher weniger geeignet. Daher kann über den Parameter **-Output GridView** ein Gridview mit den bekannten Filteroptionen anstelle des PowerShell-Objects ausgegeben werden:

Out-GridView									
Filter									P
🕂 Add criteria 👻									
DeviceID	^ IP	X-Forwarded-For	X-Forwarded-Port	X-MS-EdgelP	X-ExCompld	MS-ASProtocolVersion	X-MS-PolicyKey	X-OriginalRequestHost	X-Orig
DeviceFriendlyName									
DeviceModell									
DeviceOS	1 254	102 169 101 254	20000		ClientA scarsEcontEnd	16.1	1062221472	mail crachwark da	442.64
Status	1.2.34	192,100,101,234	30900		ClientAccessrionterio	10.1	1003221472	mail.crashwork.de	445:110
L LogEntry									
RequestTime									
ServerName	1.254	192.168.101.254	30980		ClientAccessFrontEnd	16.1	1863221472	mail.crashwork.de	443:ht
Assembly version									
Content-Length	1.254	192.168.101.254	30980		ClientAccessFrontEnd	16.1	1863221472	mail.crashwork.de	443:ht
Content-Tune	~								
Add Cancel									
			20000						
m-mx2.crashwork.global:444 192.168.	101.254	192.168.101.254	30980		ClientAccessFrontEnd	10.1	1863221472	mail.crashwork.de	443:ht

(formatierte) Tabellen-Ausgabe

Ebenso ist aber auch eine tabellarische Anzeige sinnvoll. Natürlich könnte der Befehl auch an Format-Table gepiped werden. Aber das Ausgabeobject hat sehr viele Spalten. Viele werden nicht direkt weiterhelfen. Um eine aufwändige Aussortierung zu ersparen habe ich hinter dem Parameter **-output FormatTable** eine vorgefilterte Tabelle verbaut. So gelingt der Überblick sehr einfach:

P5 C:\>>	parse-ActiveSyr	ncDebugLog -Mailbox	"tessa.te	est" -output	FormatTable				
DeviceFriendl	yName LogEntry	RequestTime	Latency	ServerName	ResponseHeader	AccessState	AccessStateReason	X-FE-ClientIP	X-OriginalRequestHost
H. C. 1		2021 10 10 18:01:26			UTTO /1 1 200 OK	Allowed	clobal	102 169 101 264	mail snashwark da
M-CL1	1	2021-10-19 18:01:20	v v	M-MX2	HTTP/1.1 200 OK	Allowed	Global	192.108.101.234	mail.crashwork.de
M-CLI	÷	2021-10-19 18:01:20	0	M-MX2	HTTP/1.1 200 OK	Allowed	GIODAI	192.108.101.234	mail. crashwork. de
M-CL1	4	2021-10-19 18:01:32	U O	M-MX2	HTTP/1.1 200 OK	Allowed	Global	192.168.101.254	mail.crashwork.de
M-CL1	3	2021-10-19 18:01:32	0	M-MX2	HTTP/1.1 200 OK	Allowed	GIODAI	192.168.101.254	mail.crashwork.de
M-CL1	4	2021-10-19 18:01:34	0	M-MX2	HTTP/1.1 200 OK	Allowed	Global	192.168.101.254	mail.crashwork.de
M-CL1	5	2021-10-19 18:01:34	0	M-MX2	HTTP/1.1 200 OK	Allowed	Global	192.168.101.254	mail.crashwork.de
M-CL1	6	2021-10-19 18:05:42	1	M-MX2	НТТР/1.1 200 ОК	Allowed	Global	192.168.101.254	mail.crashwork.de
M-CL1		2021-10-19 18:05:43	0	M-MX2	HTTP/1.1 200 OK	Allowed	Global	192.168.101.254	mail.crashwork.de
M-CL1	8	2021-10-19 18:06:10	0	M-MX2	HTTP/1.1 200 OK	Allowed	Global	192.168.101.254	mail.crashwork.de
M-CL1	9	2021-10-19 18:06:10	1	M-MX2	HTTP/1.1 200 OK	Allowed	Global	192, 168, 101, 254	mail.crashwork.de
M-CL1	10	2021-10-19 18:06:11	0	M-MX2	HTTP/1.1 200 OK	Allowed	Global	192, 168, 101, 254	mail.crashwork.de
M-CL1	11	2021-10-19 18:12:54	Ō	M-MX2	HTTP/1.1 200 OK	Allowed	Global	192, 168, 101, 254	mail.crashwork.de
M-CL1	12	2021-10-19 18:12:54	1	M-MX2	HTTP/1 1 200 OK	Allowed	Global	192 168 101 254	mail crashwork de
M-CL1	13	2021-10-19 18.17.07	ō	M-MY2	HTTP/1 1 200 OK	Allowed	Global	192 168 101 254	mail crashwork de
M-CL1	14	2021-10-19 18.17.47	ŏ	M-MY2	HTTP/1 1 200 OK	Allowed	clobal	192 168 101 254	mail crashwork de
M-CL1	10	2021 10 10 19 17 17	ě	M MY2	HTTP/1.1 200 OK	Allowed	clobal	102 169 101 254	mail crachwork do
M-CLI	10	2021-10-19 18:17:51	, North Contraction of the second sec	M-MX2	1117/1.1 200 OK	Allowed	clobal	102 168 101 254	mail chashwork.de
M-CL1	10	2021-10-19 18:17:51	Ů,	M-MXZ	HTTP/1.1 200 OK	Allowed	Global	192.108.101.254	mail. crashwork. de
M-CL1	1/	2021-10-19 18:18:10	0	M-MXZ	HTTP/1.1 200 OK	Allowed	GIODAI	192.168.101.254	mail.crashwork.de
M-CL1	18	2021-10-19 18:18:10	0	M-MX2	нттр/1.1 200 ок	Allowed	Global	192.168.101.254	mail.crashwork.de

Achtung: Diese Ausgabeform kann nicht mehr mit der Pipeline weiterverarbeitet werden!

HTML-Ausgabe

Vielleicht ist euch die Visualisierung im Browser lieber? Kein Problem: mit **-output html** wird eine html-Datei erzeugt und automatisch im Standard-Browser geöffnet:

P5 C:\>>	parse-ActiveSyncDebugLo	g -Mailbox "te	essa.test"	-outpu	t html						
P5 C:\>>	/C:/Users/ADMINI~1/AppDa	nta/Loca × +									- 🗆 X
	\leftarrow \rightarrow C \textcircled{a}	① file:///C:/User	rs/ADMINI~	1/AppDat	a/Local/T	emp/2/A	ctiveSyncDe	bugLog.htr	n	▽	☆ ॥\ ⊡ © =
	DeviceID	DeviceFriendlyName	DeviceModell	DeviceOS	Status	LogEntry	RequestTime	ServerName	AssemblyVersion	Identifier	URI
	155117C2317744E297C336FC48425F3A	M-CL1		Windows 10.0.18363	DeviceOk	0	2021-10-19 18:01:26	M-MX2	15.01.1913.003	3E58D373	POST /Microsoft-Server-ActiveSync/Proxy /default.eas?Cmd=FolderSync& DeviceId=155117C2317744E297C336FC4842 DeviceType=WindowsMail HTTP/1.1
	155117C2317744E297C336FC48425F3A	M-CL1		Windows 10.0.18363	DeviceOk	1	2021-10-19 18:01:26	M-MX2	15.01.1913.003	7EA30D4F	POST /Microsoft-Server-ActiveSync/Proxy /default.eas?Cmd=Sync& DeviceId=155117C2317744E297C336FC4842 DeviceType=WindowsMail HTTP/1.1
	155117C2317744E297C336FC48425F3A	M-CL1		Windows 10.0.18363	DeviceOk	2	2021-10-19 18:01:32	M-MX2	15.01.1913.003	23E268CC	POST /Microsoft-Server-ActiveSync/Proxy /default.eas?Cmd=FolderSync& DeviceId=155117C2317744E297C336FC4842 DeviceType=WindowsMail HTTP/1.1
	155117C2317744E297C336FC48425F3A	M-CL1		Windows 10.0.18363	DeviceOk	3	2021-10-19 18:01:32	M-MX2	15.01.1913.003	465382B9	POST /Microsoft-Server-ActiveSync/Proxy /default.eas?Cmd=Sync& DeviceId=155117C2317744E297C336FC4842 DeviceType=WindowsMail HTTP/1.1
	155117C2317744E297C336FC48425F3A	M-CL1		Windows 10.0.18363	DeviceOk	4	2021-10-19 18:01:34	M-MX2	15.01.1913.003	387AD545	POST /Microsoft-Server-ActiveSync/Proxy /default.eas?Cmd=FolderSync& DeviceId=155117C2317744E297C336FC4842 DeviceType=WindowsMail HTTP/1.1
	1551170221774E2020224E040425E24	N CL		Windows	D	6	2021-10-19		15 01 1012 002	4D 2D00ED	POST /Microsoft-Server-ActiveSync/Proxy /default.eas?Cmd=Sync&

weitere Parameter und deren Kombinationen

Natürlich ist eine Hilfe in der Funktion enthalten. Hier gibt es ausführliche Erklärungen und viele Beispiele zum Verständnis:



E	Device Next	5
Synopsis	welogs einer Mailhox auslesen und narsen.	Se
Description Mit dieser Funktion kann von einer das ActiveSyncDebuglog ausgelesen n vor. Die Funktion parst die Inform Objekt strukturiert zurück.	Mailbox für ein oder mehrere auswählbare ActiveSync-Device erden. Normalerweise liegen diese Daten in einem Fileßtext tionen aus dem Text heraus und gibt sie als PowerShell-	
Wichtig: Für das Auslesen des Logs 1) Das ActiveSyncDebugLog muss akt 2) Nach der Aktivierung muss ein Sy protokolliert wird	sind 2 Voraussetzungen zu erfüllen: viert sein. mchronisierungsversuch vom Device erfolgen, der dann	
Das DebugLog muss pro Mailbox mit	liesem Aufruf aktiviert werden:	
Set-CASMailbox -Identity "Tess	.Test" -ActiveSyncDebugLogging \$true	
Nach 72 Stunden deaktiviert der Ex der Analyse sollte es aber manuell	hange Server das Logging für die Mailbox automatisch. Nach mit Set-CASMailbox wieder deaktiviert werden.	
Parameters		
-Mailbox <string></string>		