



RBCLM durch PSMail ersetzen

RB-Commandlinemailer durch RB-PSMail ersetzen

Vorarlberger Standardschulinstallation

Autoren: Lukas Franz, Erich Vonach,
Andreas Linder

Besuchen Sie uns im Internet
<http://www.vobs.at/rb>

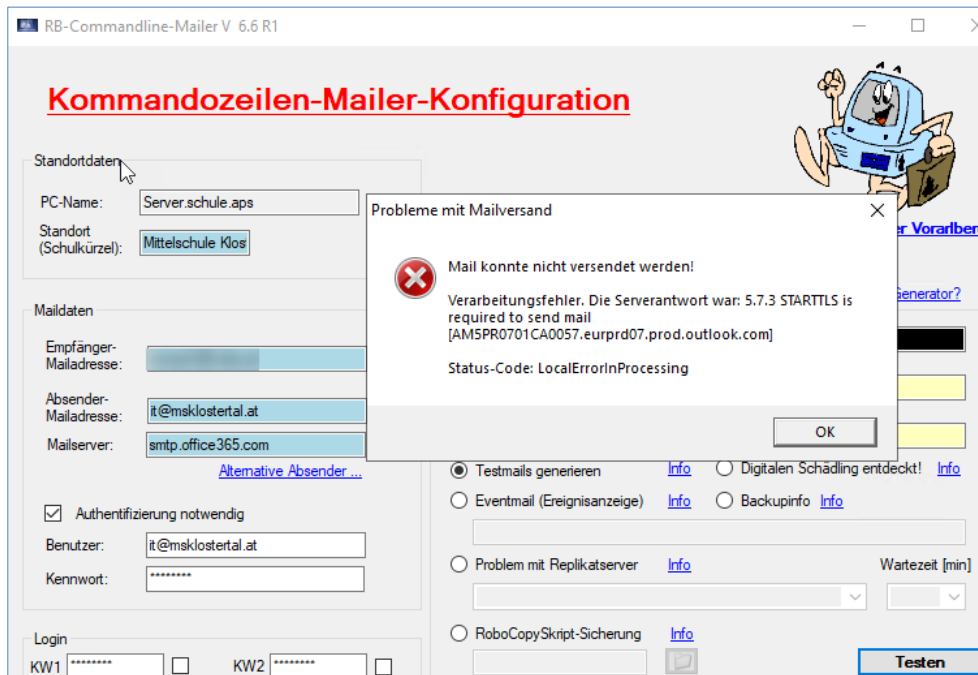
© 2022 Vorarlberger Bildungsservice
© IT-Regionalbetreuer
6900 Bregenz, Römerstraße 14
Alle Rechte vorbehalten

Inhalt

1.	Vorbemerkung	3
2.	Vorgangsweise	4
2.1.	copy_PSMail.bat	4
2.2.	Server	4
2.2.1.	Skript-Speicherorte	4
2.2.2.	Anpassungen	5
2.2.2.1.	CleanUP-WSUS.ps1	5
2.2.2.2.	autosoffload.bat	5
2.2.3.	Konfiguration von PSMail30(-mit-Attachement).ps1	5
2.2.4.	Tasks	6
2.2.4.1.	E-Mail-Alert_Ransomware	7
2.2.4.2.	E-Mail-Alert_Virus_Achtung_ab_W2k16	7
2.3.	Host 1	8
2.3.1.	Skript-Speicherorte	8
2.3.2.	Anpassungen	8
2.3.2.1.	Konfiguration von PSMail30(-mit-Attachement).ps1	8
2.3.3.	Tasks	9
2.3.3.1.	E-Mail-Alert_Serverstatusreport	9
2.3.3.2.	E-Mail-Alert_HD-Fehler1216.ps1	10
2.3.3.3.	E-Mail-Alert_HD-Fehler	10
2.3.3.4.	E-Mail-Alert_HD-Fehler4879.ps1	10
2.3.3.5.	E-Mail-Alert_HD-FehlerWarnung.ps1	10
2.3.3.6.	E-Mail-Alert_Replikationsfehler-ID-10028.ps1	11
2.3.3.7.	E-Mail-Alert_Replikationsfehler-ID-29292.ps1	11
2.3.3.8.	E-Mail-Alert_Replikationsfehler-ID-32022.ps1	11
2.3.3.9.	E-Mail-Alert_Replikationsfehler-ID-32086.ps1	11
2.3.3.10.	E-Mail-Alert_Replikationsfehler-ID-32088.ps1	11
2.3.3.11.	E-Mail-Alert_Replikationsfehler-ID-32315.ps1	11
2.3.3.12.	E-Mail-Alert_Replikationsfehler-ID-32570.ps1	11
2.3.3.13.	E-Mail-Alert_Replikationsfehler-ID-33680.ps1	11
3.	Anhang	12
3.1.	Probleme beim Mailversand	12
3.1.1.	Powershell Befehle zulassen (bei Bedarf)	12
3.1.1.1.	Kontrolle	12
3.1.2.	SMTP-Versand ermöglichen (bei Bedarf)	12
3.1.3.	Zugriff von Host1 auf die virtuellen Rechner ermöglichen (bei Bedarf)	12
3.1.4.	Fehler beim Mailversand	13
3.1.4.1.	Das Blockieren des Ausführens von Dateien aufheben	14
3.2.	Skripts	15
3.2.1.	Server	15
3.2.1.1.	E-Mail-Alert_Ransomware.ps1	15
3.2.1.2.	E-Mail-Alert_Virus-ID116.ps1	15
3.2.1.3.	autosoffload.bat	15
3.2.2.	Host1	23
3.2.2.1.	E-Mail-Alert_HD-Fehler.ps1	23
3.2.2.2.	E-Mail-Alert_HD-Fehler1216.ps1	24
3.2.2.3.	E-Mail-Alert_HD-Fehler4879.ps1	24
3.2.2.4.	E-Mail-Alert_HD-FehlerWarnung.ps1	24
3.2.2.5.	E-Mail-Alert_Replikationsfehler (ein Skript für alle Fehler)	24
3.2.2.6.	RB-PSMail_ServerStatusRep.ps1	25

1. Vorbemerkung

Das RB-Tool „Commandline-Mailer“ (Danke an den Programmator Thomas Schipflinger!), das lange Jahre dazu genutzt wurde div. Status- bzw. Warnmails vom Server aus zu verschicken, wird nicht mehr weiter entwickelt. Da es nicht mit Mailadressen umgehen kann, die zwingend eine Verschlüsselung für den SMTP-Versand benötigen (z.B. die Microsoft-365-Adressen), brauchen wir dafür einen Ersatz.



Als sehr gute Alternative bietet sich für unsere Zwecke „RB-PSMail30“ (Dank an IT-Regionalbetreuer Lukas Franz!) an.



Die vorliegende Beschreibung ersetzt nicht die Original-Anleitung, sondern zeigt, an welchen Stellen in der VlbG. Standard-Schulinstallation welche Änderungen für den Umstieg von RB-CLM auf RB-PSMail erforderlich sind.

Die erforderlichen Skripts und Tasks stehen als Download im IT-Supportbereich zur Verfügung.

2. Vorgangsweise

Download der erforderlichen [Dateien](#) z.B. nach C:\Temp auf den Server.

2.1. copy_PSMail.bat

Die Ausführung dieses Skripts ist **optional**. Die Kopie der Dateien kann natürlich auch händisch erfolgen. Durch das Ausführen werden alle erforderlichen Dateien in den vorgesehenen Ordner auf den Host1 und auf den Server kopiert.

Host1: C:_Setup\RB-PSMail
Server: D:\Install\Setup\$_Administrator\RB-PSMail
D:\CDs\Softwareverteilung_autodownload\Tools

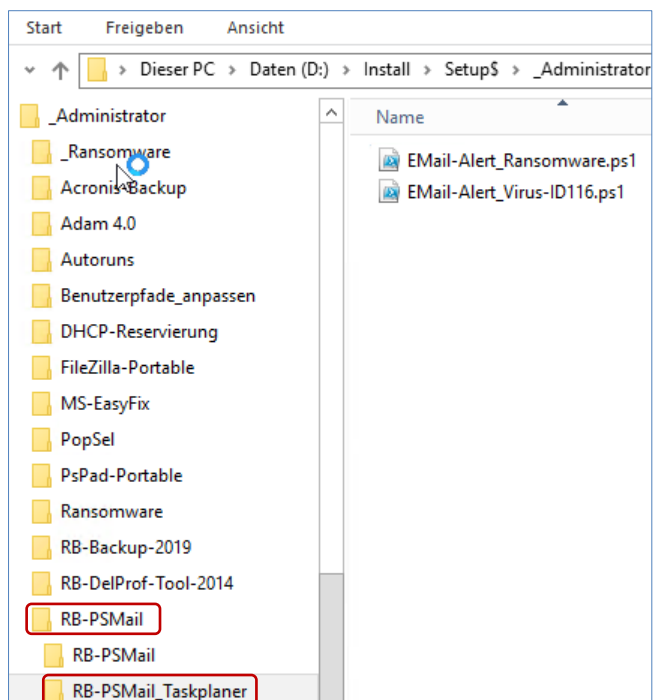
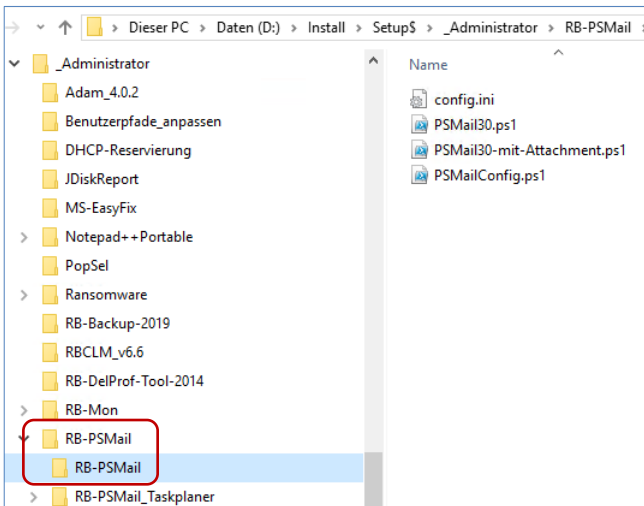
2.2. Server

2.2.1. Skript-Speicherorte

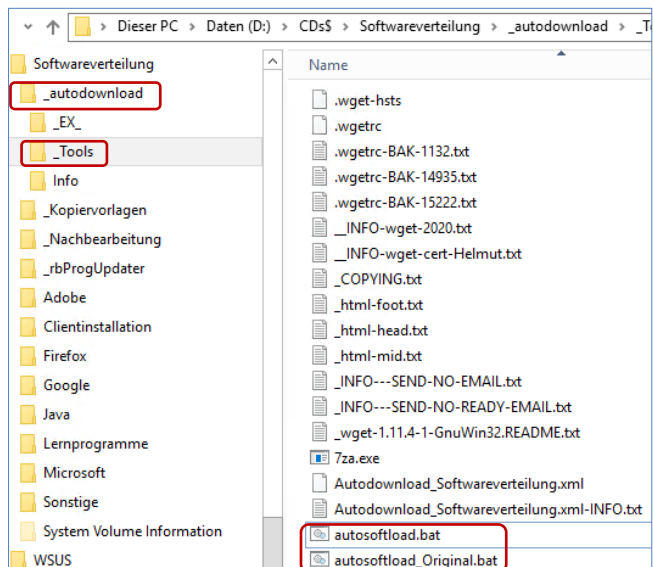
Kopie des „Programmskripts“ nach
...\setup\$_Administrator\RB-PSMail\RB-PSMail

Hinweis: Es wäre natürlich auch ein anderer Speicherort möglich, in den abhängigen Skripts wird aber auf diesen Speicherort verwiesen.

Kopie der „Taskplanerskripts“ nach
...\setup\$_Administrator\RB-PSMail\RB-PSMail_Taskplaner



Umbenennen der originalen Datei *autosoftload.bat* in *autosoftload_Original.bat*
Kopie von „autosoftload.bat“ nach
...\Softwareverteilung_autodownload\Tools



2.2.2. Anpassungen

2.2.2.1. CleanUP-WSUS.ps1

Nur der Inhalt des Skripts muss auf die neue Emailadresse umgestellt werden.
Der aufrufende Task bleibt unverändert.

"D:\Install\Setup\$_Administrator\Scripts\WSUS-Cleanup\CleanUP-WSUS.ps1"

```
569 #####
570 #####
571 Configuration Variables #
572 Simple Configuration #
573 #####
574
575 #####
576 Mail Report Setup Variables #
577 #####
578
579 From: address for email notifications (it doesn't have to be a real email address, but if you're sending through Gmail it must be
580 your Gmail address). Example: 'WSUS@domain.com' or 'email@gmail.com'
581 :ring|$AdamjMailReportEmailFromAddress = 'it@bildungscampus-lech.at'
582
583 To: address for email notifications. Example: 'firstname.lastname@domain.com'
584 :ring|$AdamjMailReportEmailToAddress = 'it@bildungscampus-lech.at'
585
586 Subject: of the results email
587 :ring|$AdamjMailReportEmailSubject = 'MSLE___WSUS Adamj-Script Cleanup Results: MS/VS Lech'
588
589 Enter your SMTP server name. Example: 'mailserver.domain.local' or 'mail.domain.com' or 'smtp.gmail.com'
590 Note Gmail Settings: smtp.gmail.com Port:587 SSL:Enabled User:user@gmail.com Password (if you use 2FA, make an app password).
591 :ring|$AdamjMailReportSMTPServer = 'smtp.office365.com'
592
593 Enter your SMTP port number. Example: '25' or '465' (Usually for SSL) or '587' or '1025'
594 :ring|$AdamjMailReportSMTPPort = '587'
595
596 Do you want to enable SSL communication for your SMTP Server?
597 :ring|$AdamjMailReportSMTPServerEnableSSL = $True
598
599 Do you need to authenticate to the server? If not, leave blank. Note: if your password includes an apostrophe, use 2 apostrophes so
600 :ring|$AdamjMailReportSMTPServerUsername = 'it@bildungscampus-lech.at'
601 :ring|$AdamjMailReportSMTPServerPassword =
602
```

2.2.2.2. autosoffload.bat

Das Skript ist eine angepasste Kopie des originalen Skripts autosoffload.bat im Ordner
"D:\CDs\$\Softwareverteilung_autodownload_Tools"

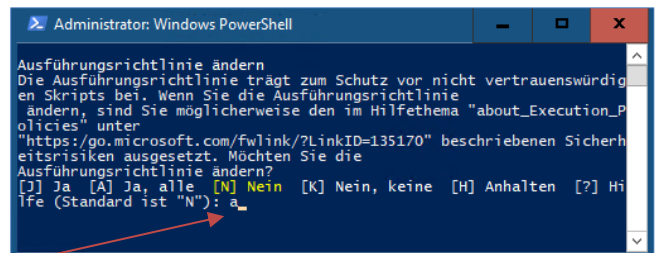
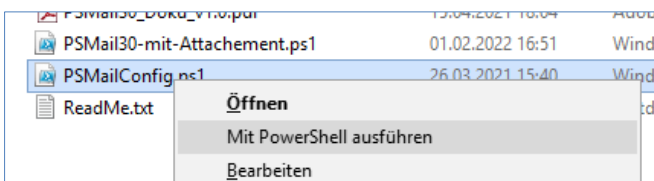
Es kann selber nach der Vorgabe im Anhang verändert werden, oder aus den heruntergeladenen Dateien übernommen werden.



Stand Juni 2022: Es steht eine neue Version des Autodownloaders (2.8) zur Verfügung. Diese Version enthält alle notwendigen Neuerungen mit Integration von PSMail. Die Datei „autosoffload.bat“ in diesem Paket wird deshalb nicht mehr automatisch in den „_autodownload“-Ordner kopiert.

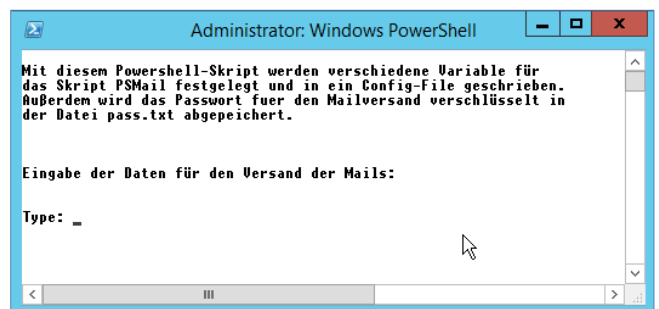
2.2.3. Konfiguration von PSMail30(-mit-Attachement).ps1

Aufruf von PSMailConfig.ps1 [RM]



Hinweise:

- Es KANN sein, dass eine Aufforderung erscheint, die Ausführungsrichtlinie zu ändern → Mit „a“ bestätigen.
- Sollte – entgegen der Originalanleitung – beim Aufruf die (unvorhergesehene) Abfrage nach dem „Typ“ erscheinen, so ist ein „f“ für „File“ einzutragen.



```

Administrator: Windows PowerShell
das Skript PSMail festgelegt und in ein Config-File geschrieben.
Das Passwort fuer den Mailversand liegt danach unverschlüsselt in diesem
File unverschlüsselt.

Eingabe der Daten für den Versand der Mails:

Schulbezeichnung: MS Klostertal
Emailadresse Absender: it@msklostertal.at
Zugehöriges Passwort für den smtp-Versand eingeben: 
SMTP-Server: smtp.office365.com
SMTP-Port (meistens 587): 587
Emailadresse Empfänger: it@msklostertal.at

Kontrolle der Daten: Wenn die Daten richtig sind, bestätigen sie mit 'j'
- es wird ein Testmail versendet.
Ansonsten beendet sich das Skript und muss erneut ausgeführt werden.

Schulbezeichnung - MS Klostertal
Emailadresse Absender - it@msklostertal.at
SMTP-Server - smtp.office365.com
SMTP-Port - 587
Emailadresse Empfänger - it@msklostertal.at

Stimmen die Daten? j/n: _

```

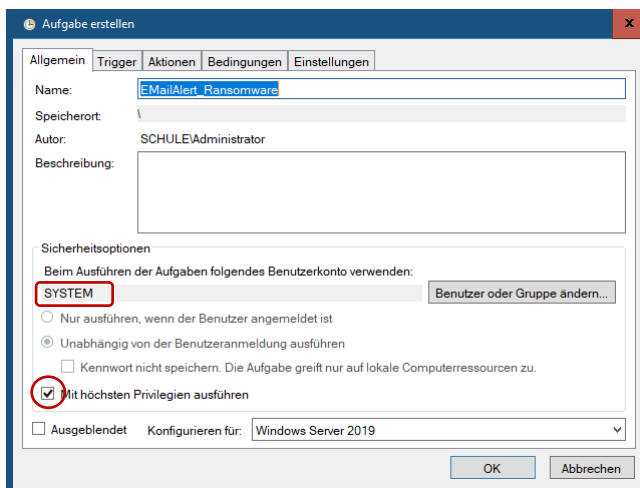
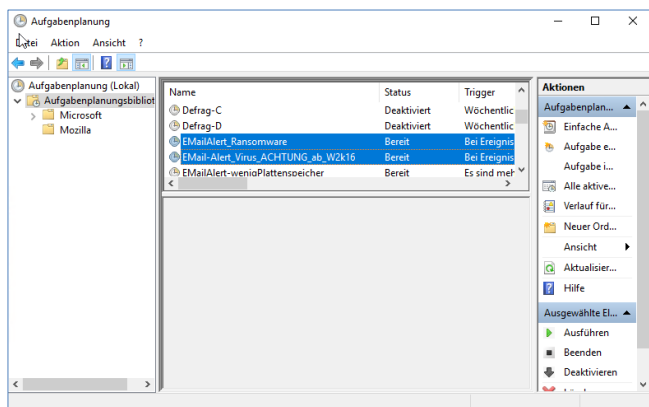
SMTP-Server und SMTP-Port hängen natürlich vom verwendeten Mailaccount ab.

Nach der Bestätigung mit „j“ werden im „Programmordner“ die Dateien `config.ini` angelegt. Sie enthält auch das Passwort im Klartext, da aber nur Administratoren Zugriff auf den Ordner haben, ist dies kein Sicherheitsproblem.

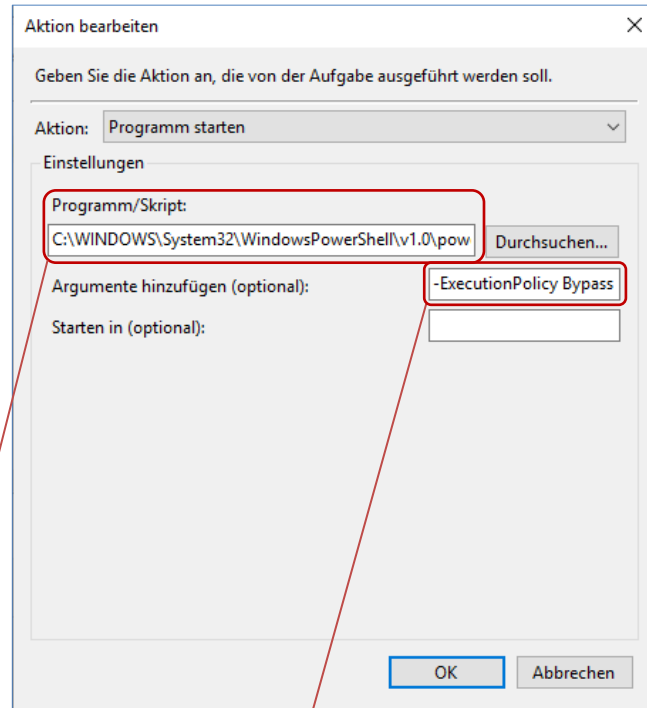
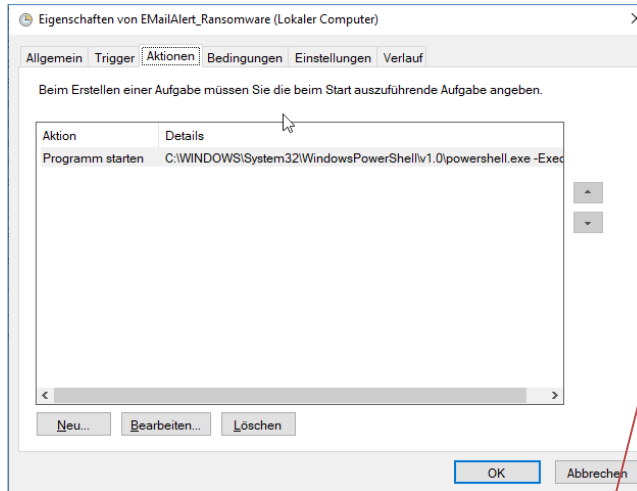
2.2.4. Tasks

Die Tasks

- „EMailAlert_Ransomware“ und „EMailAlert_Virus“ werden entweder angepasst oder gelöscht und durch die heruntergeladenen Tasks ersetzt (zu finden in `D:\Install\Setup$_Administrator\RB-PSMail\RB-PSMail_Taskplaner`)
 - Der Task „EMailAlert-wenigPlattenspeicher“ wird ersatzlos entfernt.
- Alle Tasks können mit Systemrechten ausgeführt werden.

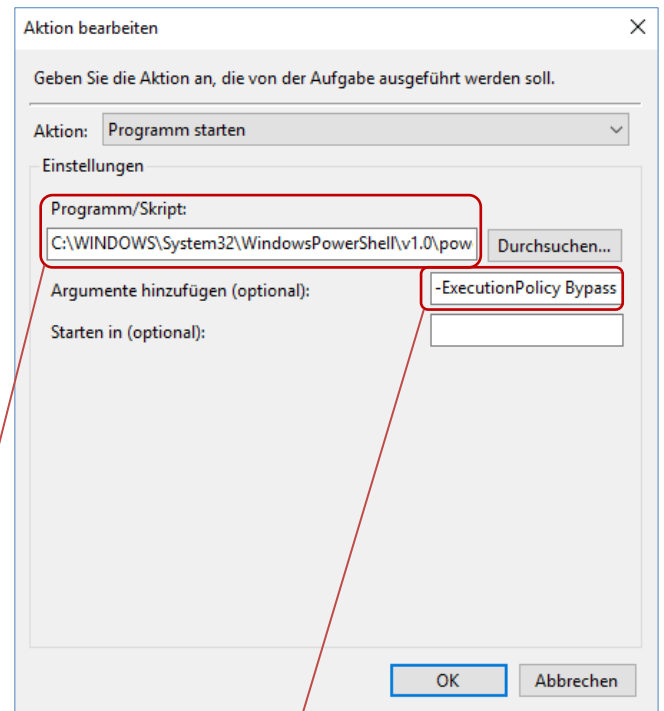
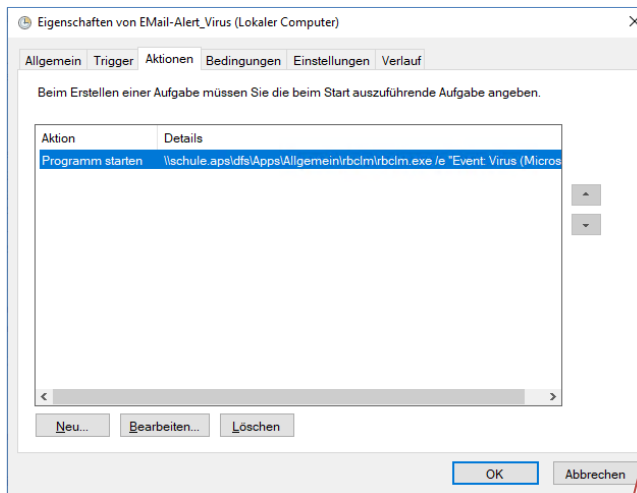


2.2.4.1. EMailAlert Ransomware



```
C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe  
-ExecutionPolicy Bypass "D:\Install\Setup$\_Administrator\RB-PSMail\RB-  
PSMail_Taskplaner\EMail-Alert_Ransomware.ps1"
```

2.2.4.2. EMailAlert Virus Achtung ab W2k16

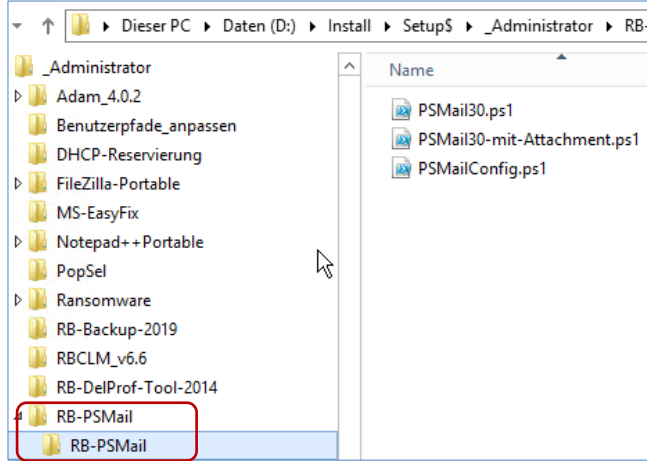


```
C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe  
-ExecutionPolicy Bypass "D:\Install\Setup$\_Administrator\RB-PSMail\RB-  
PSMail_Taskplaner\EMail-Alert_Virus-ID116.ps1"
```

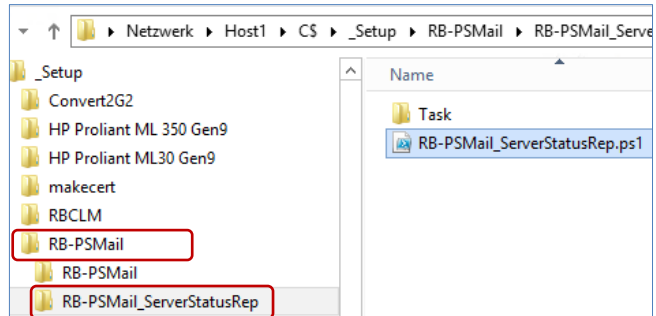

2.3. Host 1

2.3.1. Skript-Speicherorte

Kopie der „Programmskripts“ nach `C:_Setup\RB-PSMail\RB-PSMail` auf den Host 1.
Hinweis: Es wäre natürlich auch ein anderer Speicherort möglich, in den abhängigen Skripts wird aber auf diesen Speicherort verwiesen.



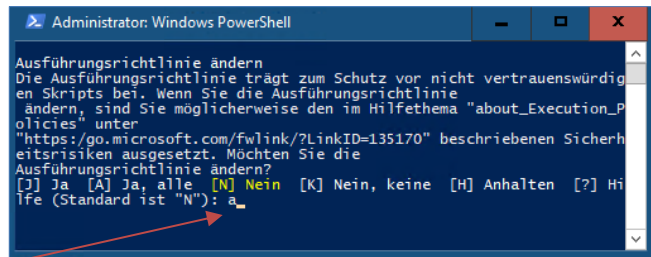
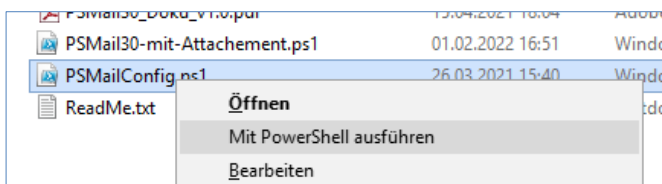
Kopie des „Serverstatusskripts“ nach `C:_Setup\RB-PSMail\RB-PSMAIL-ServerStatusRep` auf den Host 1



2.3.2. Anpassungen

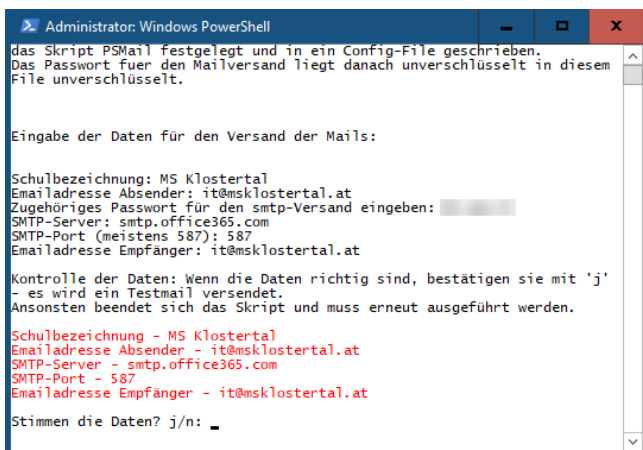
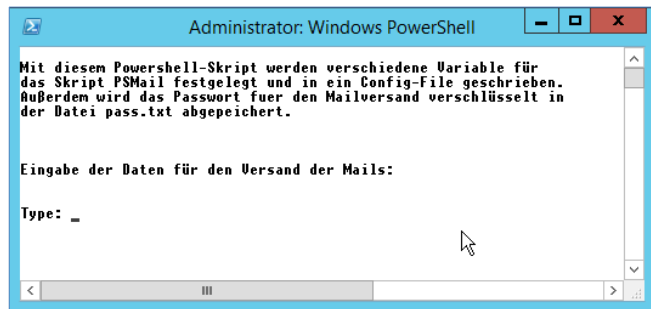
2.3.2.1. Konfiguration von PSMail30(-mit-Attachment).ps1

Aufruf von `PSMailConfig.ps1` [RM]



Hinweise:

- Es KANN sein, dass eine Aufforderung erscheint, die Ausführungsrichtlinie zu ändern → Mit „a“ bestätigen.
- Sollte – entgegen der Originalanleitung – beim Aufruf die (unvorhergesehene) Abfrage nach dem „Typ“ erscheinen, so ist ein „f“ für „File“ einzutragen.



SMTP-Server und SMTP-Port hängen natürlich vom verwendeten Mailaccount ab.

Nach der Bestätigung mit „j“ wird im „Programmordner“ die Datei `config.ini` angelegt. Sie enthält auch das Passwort im Klartext, da aber nur Administratoren Zugriff auf den Ordner haben, ist dies kein Sicherheitsproblem.

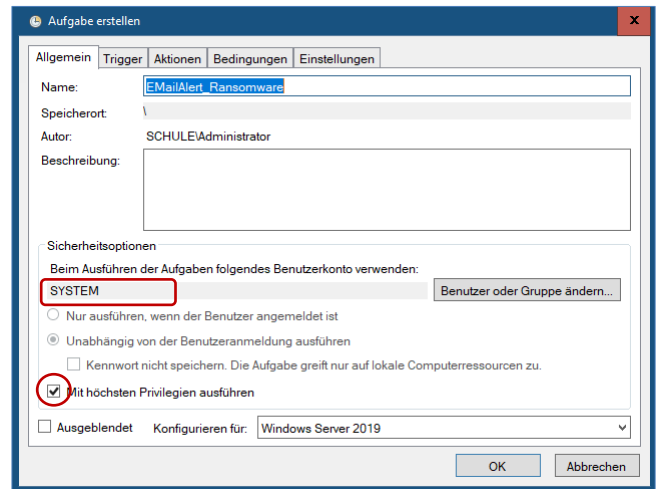
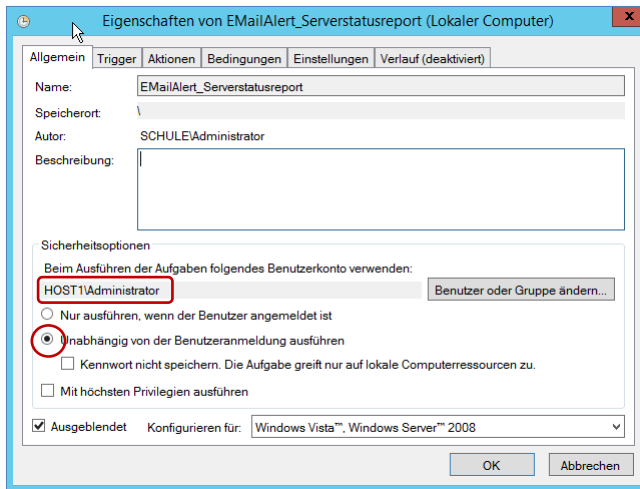
Die Datei kann händisch bearbeitet werden, es dürfen aber keinesfalls Zeilen vertauscht werden!

2.3.3. Tasks

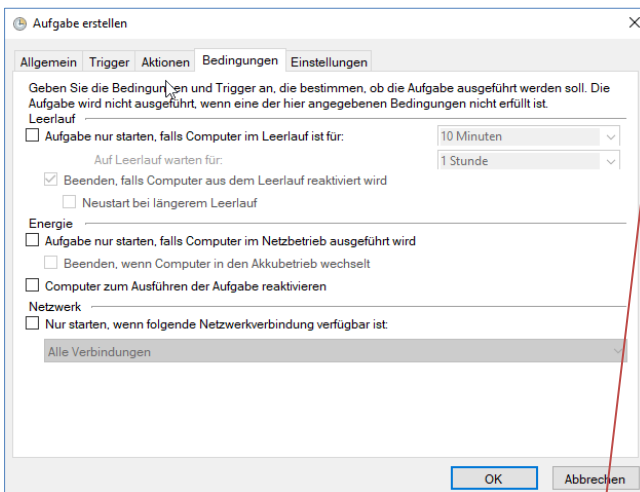
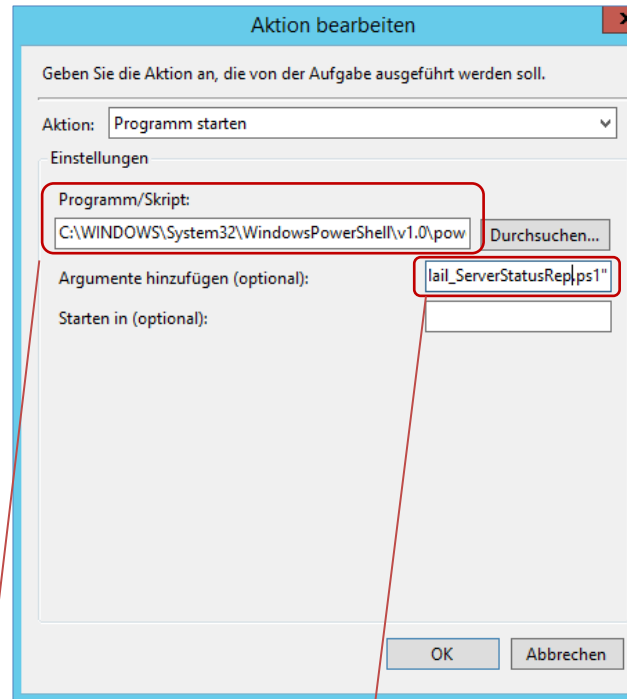
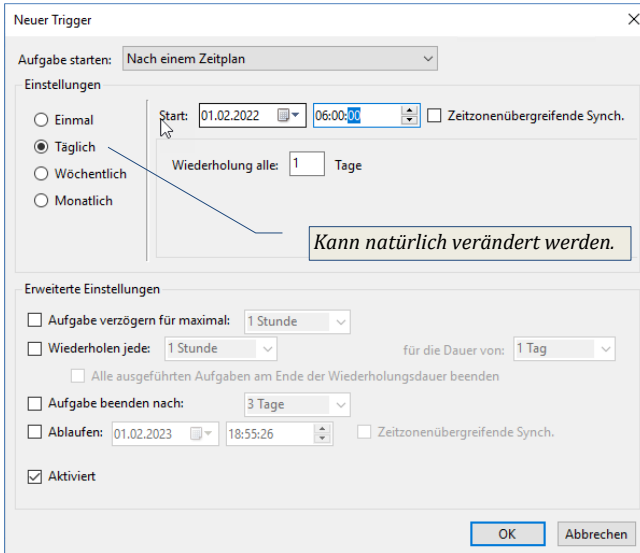
Die vorhandenen Tasks werden entweder angepasst oder gelöscht und durch die heruntergeladenen Tasks ersetzt (zu finden in `C:_Setup\RB-PSMail\RB-PSMail_Taskplaner\Tasks`).

Der Task `E-MailAlert_Serverstatusreport` MUSS mit Administratorenrechten ausgeführt werden

Alle anderen Tasks können mit Systemrechten ausgeführt werden.




2.3.3.1. E-MailAlert_Serverstatusreport



```
C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe
-ExecutionPolicy Bypass "C:\_Setup\RB-PSMail\RB-PSMail_ServerStatusRep\RB-PSMail_ServerStatusRep.ps1"
```

2.3.3.2. [EMail-Alert HD-Fehler1216.ps1](#)

```
C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe  
-ExecutionPolicy Bypass "C:\_Setup\RB-PSMail\RB-PSMail_Taskplaner\EMail-Alert_HD-  
Fehler1216.ps1"
```

 it@msklostertal.at | it@msklostertal.at
Schulbezeichnung Event HD-defekt (Fehler ID 1216)


Dieses Mail wurde vom HOST1 der Schulbezeichnung gesendet.
Ereigniszeitpunkt: 02/02/2022 14:09:52

Storage Agents vom HOST1 hat ein Problem festgestellt.

Versandt mit PSMail30.ps1

2.3.3.3. [EMail-Alert HD-Fehler](#)

```
C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe  
-ExecutionPolicy Bypass "C:\_Setup\RB-PSMail\RB-PSMail_Taskplaner\EMail-Alert_HD-  
Fehler.ps1"
```

 it@msklostertal.at | it@msklostertal.at
Schulbezeichnung Event HD-defekt (Fehler ID 1200)


Dieses Mail wurde vom HOST1 der Schulbezeichnung gesendet.
Ereigniszeitpunkt: 02/02/2022 14:09:57

Storage Agents vom HOST1 hat ein Problem festgestellt.

Versandt mit PSMail30.ps1

2.3.3.4. [EMail-Alert HD-Fehler4879.ps1](#)

```
C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe  
-ExecutionPolicy Bypass "C:\_Setup\RB-PSMail\RB-PSMail_Taskplaner\EMail-Alert_HD-  
Fehler4879.ps1"
```

 it@msklostertal.at | it@msklostertal.at
Schulbezeichnung Event HD-defekt (Fehler ID 4879)


Dieses Mail wurde vom HOST1 der Schulbezeichnung gesendet.
Ereigniszeitpunkt: 02/02/2022 14:09:54

Agentless Mangement Service vom HOST1 hat ein Problem festgestellt.

Versandt mit PSMail30.ps1

2.3.3.5. [EMail-Alert HD-FehlerWarnung.ps1](#)

```
C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe  
-ExecutionPolicy Bypass "C:\_Setup\RB-PSMail\RB-PSMail_Taskplaner\EMail-Alert_HD-  
FehlerWarnung.ps1"
```

 it@msklostertal.at | it@msklostertal.at
Schulbezeichnung Event HD-defekt (Smart Array Event 24595)

Dieses Mail wurde vom HOST1 der Schulbezeichnung gesendet.
Ereigniszeitpunkt: 02/02/2022 14:09:50

Smart Array vom HOST1 hat ein Problem festgestellt.

Versandt mit PSMail30.ps1

[2.3.3.6. EMail-Alert Replikationsfehler-ID-10028.ps1](#)

C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe
-ExecutionPolicy Bypass "C:_Setup\RB-PSMail\RB-PSMail_Taskplaner\EMail-Alert_Replikationsfehler.ps1"

[2.3.3.7. EMail-Alert Replikationsfehler-ID-29292.ps1](#)

C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe
-ExecutionPolicy Bypass "C:_Setup\RB-PSMail\RB-PSMail_Taskplaner\EMail-Alert_Replikationsfehler.ps1"

[2.3.3.8. EMail-Alert Replikationsfehler-ID-32022.ps1](#)

C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe
-ExecutionPolicy Bypass "C:_Setup\RB-PSMail\RB-PSMail_Taskplaner\EMail-Alert_Replikationsfehler.ps1"

[2.3.3.9. EMail-Alert Replikationsfehler-ID-32086.ps1](#)

C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe
-ExecutionPolicy Bypass "C:_Setup\RB-PSMail\RB-PSMail_Taskplaner\EMail-Alert_Replikationsfehler.ps1"

C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe
-ExecutionPolicy Bypass "C:_Setup\RB-PSMail\RB-PSMail_Taskplaner\EMail-Alert_Replikationsfehler.ps1"

[2.3.3.10. EMail-Alert Replikationsfehler-ID-32088.ps1](#)

C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe
-ExecutionPolicy Bypass "C:_Setup\RB-PSMail\RB-PSMail_Taskplaner\EMail-Alert_Replikationsfehler.ps1"

[2.3.3.11. EMail-Alert Replikationsfehler-ID-32315.ps1](#)

C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe
-ExecutionPolicy Bypass "C:_Setup\RB-PSMail\RB-PSMail_Taskplaner\EMail-Alert_Replikationsfehler.ps1"

[2.3.3.12. EMail-Alert Replikationsfehler-ID-32570.ps1](#)

C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe
-ExecutionPolicy Bypass "C:_Setup\RB-PSMail\RB-PSMail_Taskplaner\EMail-Alert_Replikationsfehler.ps1"

[2.3.3.13. EMail-Alert Replikationsfehler-ID-33680.ps1](#)

C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe
-ExecutionPolicy Bypass "C:_Setup\RB-PSMail\RB-PSMail_Taskplaner\EMail-Alert_Replikationsfehler.ps1"

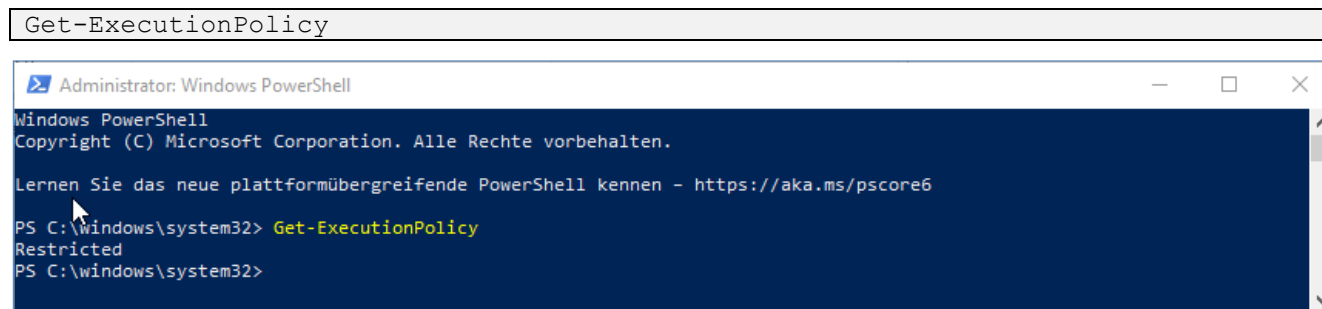
3. Anhang

3.1. Probleme beim Mailversand

3.1.1. Powershell Befehle zulassen (bei Bedarf)

3.1.1.1. Kontrolle

```
Get-ExecutionPolicy
```



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Alle Rechte vorbehalten.

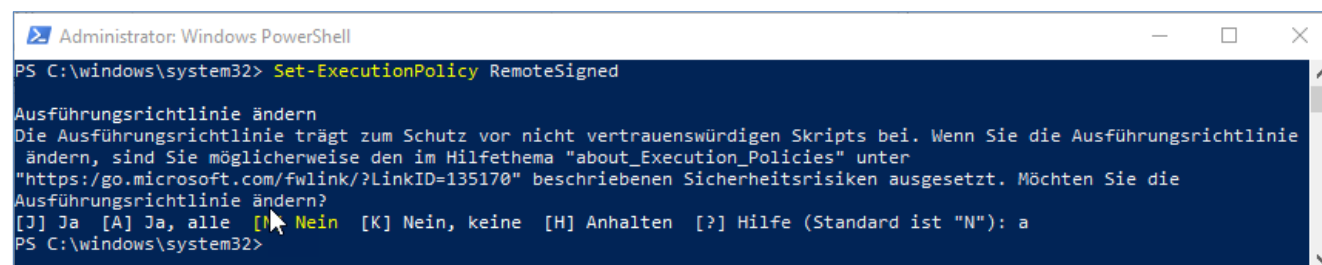
Lernen Sie das neue plattformübergreifende PowerShell kennen - https://aka.ms/pscore6

PS C:\windows\system32> Get-ExecutionPolicy
Restricted
PS C:\windows\system32>
```

Ist das Ergebnis „Restricted“

```
Set-ExecutionPolicy RemoteSigned
```

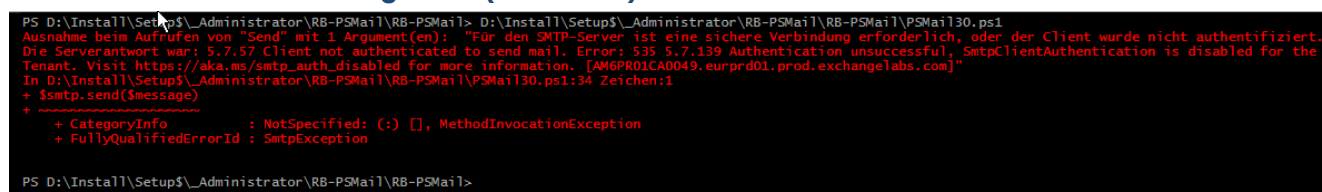
„a“ für alle



```
Administrator: Windows PowerShell
PS C:\windows\system32> Set-ExecutionPolicy RemoteSigned

Ausführungsrichtlinie ändern
Die Ausführungsrichtlinie trägt zum Schutz vor nicht vertrauenswürdigen Skripts bei. Wenn Sie die Ausführungsrichtlinie ändern, sind Sie möglicherweise den im Hilfethema "about_Execution_Policies" unter "https://go.microsoft.com/fwlink/?LinkID=135170" beschriebenen Sicherheitsrisiken ausgesetzt. Möchten Sie die Ausführungsrichtlinie ändern?
[J] Ja [A] Ja, alle [N] Nein [K] Nein, keine [H] Anhalten [?] Hilfe (Standard ist "N"): a
PS C:\windows\system32>
```

3.1.2. SMTP-Versand ermöglichen (bei Bedarf)



```
PS D:\Install\Setup$\Administrator\RB-PSMail\RB-PSMail> D:\Install\Setup$\Administrator\RB-PSMail\RB-PSMail\PSMail30.ps1
Ausnahme beim Aufrufen von "Send" mit 1 Argument(en): "Für den SMTP-Server ist eine sichere Verbindung erforderlich, oder der Client wurde nicht authentifiziert. Die Serverantwort war: 5.7.57 Client not authenticated to send mail. Error: 535 5.7.139 Authentication unsuccessful; SmtplibClientAuthentication is disabled for the Tenant. Visit https://aka.ms/smtp_auth_disabled for more information. [AM6PR01CA0049.eurprd01.prod.exchangeabs.com]"
In D:\Install\Setup$\Administrator\RB-PSMail\RB-PSMail\PSMail30.ps1:34 Zeichen:1
+ $smtp.send($message)
+ ~~~~~
+ CategoryInfo          : NotSpecified: (:) [], MethodInvocationException
+ FullyQualifiedErrorId : SmtpException

PS D:\Install\Setup$\Administrator\RB-PSMail\RB-PSMail>
```

Für den Versand von Mails muss zwingend der SMTP-Versand für die M365-Mailadresse am Tenant erlaubt werden.

Siehe [diese Anleitung](#) (Microsoft-365-SMTP-Authentication_v22.x.pdf)

3.1.3. Zugriff von Host1 auf die virtuellen Rechner ermöglichen (bei Bedarf)

Möglicherweise tritt bei der Ausführung des Skripts das Problem auf, dass vom Host1 auf Grund fehlender Berechtigungen nicht auf die virtuellen Maschinen Server und DCSchule zugegriffen werden kann.



```
il_ServerStatusRep_v2.0.ps1
ert (Ausnahme von HRESULT: 0x80070005 (E_ACCESSDENIED))
atusRep_v2.0.ps1:35 Zeichen:10
logicaldisk -Computer Server -filter "na ...

otSpecified: (:) [Get-WmiObject], UnauthorizedAccessException
ystem.UnauthorizedAccessException,Microsoft.PowerShell.Commands.GetWmiObjectCommand

ert (Ausnahme von HRESULT: 0x80070005 (E_ACCESSDENIED))
atusRep_v2.0.ps1:39 Zeichen:10
logicaldisk -Computer Server -filter "na ...

otSpecified: (:) [Get-WmiObject], UnauthorizedAccessException
ystem.UnauthorizedAccessException,Microsoft.PowerShell.Commands.GetWmiObjectCommand

ngs-Manager auf dem Computer server kann nicht geöffnet werden. Möglicherweise verfügen Sie nicht über die Be
atusRep_v2.0.ps1:58 Zeichen:9
-computername server
```

Es liegt – logischerweise – ein Rechteproblem vor. Wir umgehen dies, indem wir in der Windows-Anmeldeinformationsverwaltung auf dem Host1 die entsprechenden Informationen für die beiden



virtuellen Maschinen speichern. Dies geht am einfachsten, indem man über ein Explorerfenster einmalig z.B. auf Laufwerk C: des Servers zugreift und die Domänen-Admin Credentials eingibt. Da es sich um Domänen-Berechtigungen handelt, muss dies nur einmalig gemacht werden. Nun sollte das PS-Skript fehlerfrei durchlaufen und im Mail die entsprechenden Informationen enthalten sein.

3.1.4. Fehler beim Mailversand

Tritt dieser Fehler auf, kann es sein, dass die „SSL Cipher-Suites“ des ServicePointManagers blockieren.

```
PS C:\_Setup\RB-PSMail\RB-PSMail> C:\_Setup\RB-PSMail\RB-PSMail_Taskplaner\EMail-Alert_Replikationsfehler.ps1
Ausnahme beim Aufrufen von "Send" mit 1 Argument(en): "Fehler beim Senden von Mail."
In C:\_Setup\RB-PSMail\RB-PSMail\PSMail30.ps1:34 Zeichen:1
+ $smtp.send($message)
+ ~~~~~
+ CategoryInfo          : NotSpecified: (:) [], MethodInvocationException
+ FullyQualifiedErrorId : SmtplibException

PS C:\_Setup\RB-PSMail\RB-PSMail>
```

Lösung: „Freigabe“ der Ciphers im Skript

- nur für die verwendeten Ciphers (Tls12 reicht vermutlich aus)
[System.Net.ServicePointManager]::SecurityProtocol = 'Tls,Tls11,Tls12'
- alle Ciphers
[System.Net.ServicePointManager]::SecurityProtocol =
[System.Net.SecurityProtocolType]::GetNames([System.Net.SecurityProtocolType])

Ergänzung in **PsMail30.ps1** NUR BEI BEDARF

```
#Definition der Parameter, welche übergeben werden
Param(
    [string]$EmailSubject,
    [string]$EmailBody,
    [string]$File
)

[System.Net.ServicePointManager]::SecurityProtocol = 'Tls12'

#Variable aus config.ini importieren
$config = get-content .\config.ini

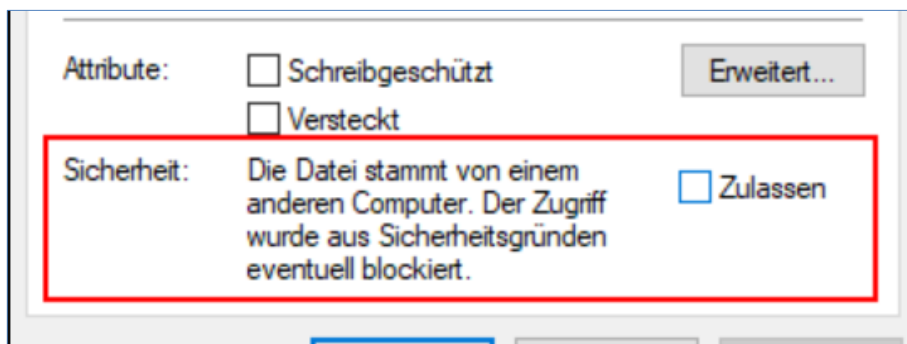
$SMTPServer = $config[2]
$SMTPPort = $config[3]
$EmailUser = $config[0]
$EmailFrom = $config[0]
$Password = Get-Content .\pass.txt | ConvertTo-SecureString
$EmailTo = $config[1]
#$EmailSubject = "BeispielSubject - könnte hier auch manuell eingetragen werden"
#$EmailBody = "BeispielBody - könnte hier auch manuell eingetragen werden"

#wird ein Anhang mitgesendet, müssen die entsprechenden Zeilen auskommentiert
#und allenfalls der richtige Parameter übergeben werden
#$att = new-object Net.Mail.Attachment($File)

$message = New-Object System.Net.Mail.MailMessage
$message.subject = $EmailSubject
$message.IsBodyHTML = $True
$message.Body = $EmailBody
$message.to.add($Emailto)
$message.from = $EmailFrom
#$message.Attachments.Add($att)
$smtp = New-Object Net.Mail.SmtpClient($SMTPServer, $SMTPPort);
$smtp.EnableSSL = $true
$smtp.Credentials = New-Object System.Net.NetworkCredential($EmailUser, $Password);
$smtp.send($message)
```

3.1.4.1. Das Blockieren des Ausführens von Dateien aufheben

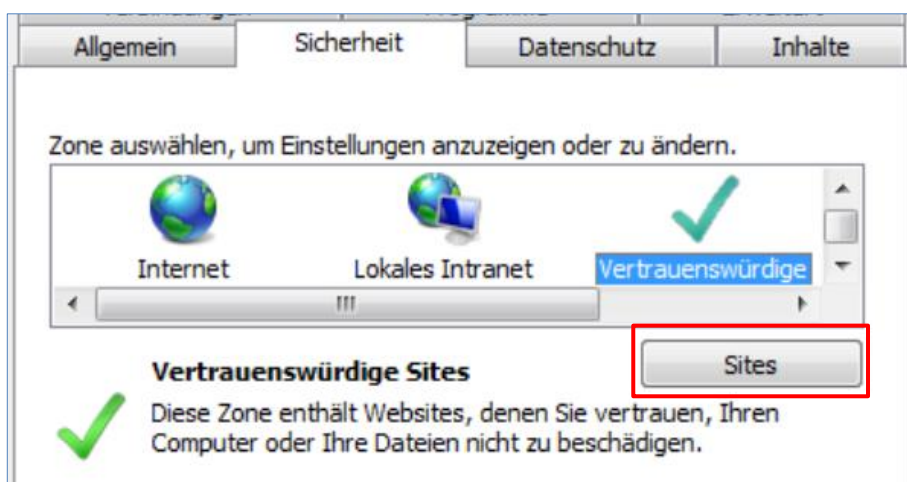
Werden nach dem Download einer ZIP-Datei die Eigenschaften dieser angezeigt, so erscheint möglicherweise ein Block „Sicherheit“ mit folgendem Hinweis: „Die Datei stammt von einem anderen Computer. Der Zugriff wurde aus Sicherheitsgründen eventuell blockiert.“



Ein Klick auf „Zulassen“ lässt diese Sicherheitswarnung verschwinden. Macht man das nicht, wirkt es sich auf alle im ZIP-File enthaltenen ausführbaren Dateien aus.

Hier würde das bedeuten, dass die PSMail-PowerShell-Skripts nicht ausgeführt werden können, wenn sie im Taskplaner eingebaut sind. Und genau das ist bei den meisten ja vorgesehen.

Eine andere Möglichkeit, das Setzen dieses Sicherheits-Attributs vorab zu unterbinden ist, die Downloadseite <https://www.vobs.at> den Vertrauenswürdigen Sites in den Internetoptionen hinzuzufügen: Über Systemsteuerung – Internetoptionen wählen wir den Reiter Sicherheit.



Über Sites können URLs hinzugefügt werden, bei denen am Ende eines Downloads keine Sicherheitsüberprüfung ausgeführt wird. Damit entfällt für alle vom VOBS heruntergeladenen Dateien der oben beschriebene Klick auf „Zulassen“.

Dies ist unabhängig davon, ob Dateien über Internetexplorer oder Chrome heruntergeladen werden.

3.2. Skripts

3.2.1. Server

3.2.1.1. [EMail-Alert Ransomware.ps1](#)

```
#Dieses Skript funktioniert nur zusammen mit dem Email-Skript PSMail30
#Variable für Mail-Betreff
$DatumZeit = Get-Date
$Sender = $env:COMPUTERNAME
$Betreff = "!! Möglicherweise Ransomware - Angriff !!"

#Email Body zusammenfügen
$Body0 = "Dieses Mail wurde vom $Sender gesendet."
$Body1 = "Ereigniszeitpunkt: $DatumZeit"
$Body2 = ""
$Body3 = "Auf dem virtuellen $Sender gibt es möglicherweise einen Ransomware-Angriff."
$Body4 = ""
$Body5 = "Versandt mit PSMail30.ps1"

$Body = "$Body0<br>$Body1<br>$Body2<br>$Body3<br>$Body4<br><br>$Body5"

#Email senden
Set-Location D:\Install\Setup$\_Administrator\RB-PSMail\RB-PSMail
.\PSMail30.ps1 -EmailSubject $Betreff -EmailBody $Body
```

3.2.1.2. [EMail-Alert Virus-ID116.ps1](#)

```
#Dieses Skript funktioniert nur zusammen mit dem Email-Skript PSMail30
#Variable für Mail-Betreff
$DatumZeit = Get-Date
$Sender = $env:COMPUTERNAME
$Betreff = "Event: Virus (Microsoft-Antimaleware ID 116)"
#$Betreff = "!! Möglicherweise Ransomware - Angriff !!"
#$Betreff = "Event: wenig Plattenspeicher"

#Email Body zusammenfügen
$Body0 = "Dieses Mail wurde vom $Sender gesendet."
$Body1 = "Ereigniszeitpunkt: $DatumZeit"
$Body2 = ""
$Body3 = "Der windows Defender hat auf dem virtuellen $Sender ein Problem festgestellt."
$Body4 = ""
$Body5 = "Versandt mit PSMail30.ps1"

$Body = "$Body0<br>$Body1<br>$Body2<br>$Body3<br>$Body4<br><br>$Body5"

#Email senden
Set-Location D:\Install\Setup$\_Administrator\RB-PSMail\RB-PSMail
.\PSMail30.ps1 -EmailSubject "$Betreff" -EmailBody $Body
```

3.2.1.3. [autosoftload.bat](#)

```
@echo off
::
set VER=v.2.6
TITLE AUTOMATISIERTER DOWNLOAD fuer Softwareverteilung
::
:: *****
:: ** AUTOSOFTLOAD.CMD **
:: **
:: *****
:: *
:: * Programmname: autosoftload.cmd *
:: *
:: * Beschreibung: Automatisierter Download der VoBS-MSI-Pakete *
:: * zur Installation ueber Softwareverteilung *
:: *
:: * Version: v.2.0 *
:: * v.2.2 FIX: holt neuere Datei nicht: --no-cache --no-proxy *
:: * v.2.3 Informiert via rbclm.exe per Infomail ueber Download *
:: * v.2.4 FIX: Neue Tools fuer HTTPS wegen VoBS-Umstellung *
:: * v.2.5 neu: RBCLM-CommandlineMailer als Variable *
:: * v.2.6 neu: Infoliste mit "neuen Downloads". RBCLM sendet Liste *
:: * v.2.7 Änderung: Mail wird mit PSMail anstatt RBCLM verschickt *
:: * Lukas Franz / Erich Vonach *
:: *
:: * Datum: 2013-09-06 *
:: * Datum: 2015-03-06 *
:: * Datum: 2016-12-06 *
:: * Datum: 2017-04-01 *
:: * Datum: 2017-05-31 *
:: * Datum: 2018-02-21 *
:: *
:: * Author: Helmut Pajnik *
:: * IT-Regionalbetreuer Bregenz Nord *
:: * mail: pajnik@vobs.at *
```



```

:: *                               Tel : 0664-6255684                               *
:: *
:: *****
::
:: *****
:: **   STARTMELDUNGENEN   **
:: **
:: *****
::
color F0
cls
echo =====
echo AUTOMATISIERTER DOWNLOAD fuer Softwareverteilung   %VER% -   H. Pajnik
echo =====
echo.
::
:: *****
:: -----
:: INSTALLATION
:: -----
::
:: siehe _README.txt
::
:: -----
:: Wichtig:
:: -----
:: Das Verzeichnis _AUTODOWNLOAD mit Inhalt (Skript und Unterverzeichnis Tools)
:: sollte im Ordner Softwareverteilung liegen,
:: z.B: \\server\Softwareverteilung\_AUTODOWNLOAD\
:: Bei anderem Pfad "set SOFTDEPLOYPATH=\\server\Softwareverteilung" anpassen!
::
:: Weitere Info in: _README_.txt
::
:: Das Skript wurde sorgfaeltig getestet,
:: der Author uebernimmt jedoch keinerlei Haftung
:: fuer eventuelle Probleme oder Datenverlust.
::
:: Helmut Pajnik
::
:: *****
::
:: *****
:: **   VERZEICHNISSE PRUEFEN - VARIABLE SETZEN   **
:: **
:: *****
::
:: Aktuelles Verzeichnis setzen
pushd %~dp0..
::           Verzeichnis, in dem die Batch aufgerufen wurde
::           2x ohne backslash ! ... eig: %~dp0\..\
set STARTDIR=%CD%
::
:: -----Pfad korrigieren-----
if "%startdir:~-1%"=="\" set startdir=%startdir:~0,-1%
:: Backslash am Ende wegschneiden
:: -----
set SOFTDIR=%STARTDIR%
set WORKDIR=%SOFTDIR%\_Tools
set LOGFILE=%WORKDIR%\AUTOSOFTupdlog.txt
set LOGERROR=%WORKDIR%\AUTOSOFTlog-Error.txt
set SOFTINFOFILENAME=_AUTOSOFTLISTAKT.txt
set SOFTINFOFILEPATH=%SOFTDIR%\%SOFTINFOFILENAME%
set SOFTINFO-TMP=%TEMP%\tmp-autodownload.txt
set SOFTINFO2-TMP=%TEMP%\tmp2-autodownload.txt
set WGET-COUNTINFO-TMP=%TEMP%\tmp-wget-countinfo.txt
set PROGINFO=%SOFTDIR%\_AKTUELLE_PROGRAMME_.txt
set PROGINFOHTML=%SOFTDIR%\_AKTUELLE_PROGRAMME_.html
set SOFTINFO-PART=%TEMP%\tmp-infopart.txt
:: ----- Anpassen -----
:: -----
set PATH-TO-MAIL-PS1=\\schule.aps\dfs\Setup$\_Administrator\RB-PSMail\RB-PSMail\PSMail30-mit-Attachment.ps1
rem set SCHULE="Schulbezeichnung" - nicht mehr notwendig
:: ----- Anpassen END -----
:: -----

```

```

set LOGNEWFILES=%WORKDIR%\AUTOSOFTlog_NEWFILES.txt
set LOGNEWFILES-PART=%WORKDIR%\AUTOSOFT_NEWFILES-PART.txt

set READYFILES-TEMP=%TEMP%\tmp-DirReadyFiles.txt
set READYINFO-PART=%TEMP%\READYINFO-PART.txt
set WORKTODO=NO

set
VOBSPATH=http://www.vobs.at/fileadmin/user_upload/itservice/downloads/standardsoftware/softwareverte
ilung
set VOBSINFOFILEPATH=%VOBSPATH%/%SOFTINFOFILENAME%

:-----
REM ===SOFTDEPLOYPATH ermitteln oder Meldung===
set SOFTDEPLOYPATH=UNKNOWN
REM Hier Pfad fuer Softwareverteilung aendern
REM z.B: set SOFTDEPLOYPATH=\\schule.aps\dfs\Softwareverteilung
REM Hier Pfad fuer Softwareverteilung aendern
REM Fuer unsere Schulumgebung wird der Pfad nun selbst generiert
REM ----
if exist "\\schule.aps\dfs\Softwareverteilung\AUTODOWNLOAD\" goto SCHULE_DFS
if exist "\\server\Softwareverteilung\AUTODOWNLOAD\" goto SCHULE_SERVER
if %SOFTDEPLOYPATH%==UNKNOWN goto WARNUNG1
if exist "%SOFTDEPLOYPATH%\" goto SOFTDEPLOYPATH_DONE
:
:WARNUNG1
echo -----FEHLER-----
echo unbekannter Pfad fuer Softwareverteilung
echo "SOFTDEPLOYPATH" ist: "%SOFTDEPLOYPATH%"
echo.
echo Bitte nach folgendem String suchen
echo "Hier Pfad fuer Softwareverteilung aendern"
echo und den Pfad laut Beispiel bei Fundstelle eintragen.
goto GANZ_ENDE
:
REM -Schulstandard APS automatisch eintragen---
: SCHULE_DFS
set SOFTDEPLOYPATH=\\schule.aps\dfs\Softwareverteilung
goto SOFTDEPLOYPATH_DONE
: SCHULE_SERVER
set SOFTDEPLOYPATH=\\server\Softwareverteilung
:SOFTDEPLOYPATH_DONE
:-----

:-----Zertifikatverzeichnis pruefen----
if exist "C:\Program Files\wget\etc\wgetrc" goto :CERT-DONE
REM xcopy "%SourceDir%\Prog-etc\*" "%ProgramFiles%\wget\etc\*" /kreischy
REM geht ins falsche Verzeichnis x86
:
echo Kopiere Dateien fuer Zertifikat nach "C:\Program Files\wget\etc\"
xcopy "%WORKDIR%\Prog-etc\*" "C:\Program Files\wget\etc\*" /kreischyq
echo - - -
REM nur fuer neuere wget version notwendig
REM certutil -addstore "AuthRoot" "thawtePrimaryRootCA-G3.crt"
:CERT-DONE
:-----Zertifikatverzeichnis pruefen-END---

cd %SOFTDIR%

echo -----
echo Hole aktuelle Infodatei von VoBS
echo -----

echo ----Log-File---Autodownload----- > "%LOGFILE%"
echo %DATE% - %TIME% >> "%LOGFILE%"
rem echo "%WORKDIR%\wget" -a "%LOGFILE%" "%VOBSINFOFILEPATH%" -N --progress=dot:binary --no-cache --
no-proxy >> "%LOGFILE%"
:
2> "%LOGERROR%" "%WORKDIR%\wget" -a "%LOGFILE%" "%VOBSINFOFILEPATH%" -N --progress=dot:binary --no-
cache --no-proxy
if "%ERRORLEVEL%"=="0" goto :NO-ERROR1

color CF
echo === FEHLER beim Download=====
REM type "%LOGERROR%"
echo siehe Logfile:
echo "%LOGFILE%"
echo =====[6 Sek]===
echo === FEHLER beim Download=== >> "%LOGFILE%"

```

```

type "%LOGERROR%" >> "%LOGFILE%"
ping -n 6 127.0.0.1 > nul
:NO-ERROR1
::
TITLE AUTOMATISIERTER DOWNLOAD fuer Softwareverteilung
type "%LOGFILE%" |find "Server file no newer than local file `"%SOFTINFOFILENAME%"` -- not retrieving"
> nul
set NEUGEHOLT=%ERRORLEVEL%
REM 0 wenn Datei schon da war, 1 wenn neu geholt
if "%NEUGEHOLT%"=="1" echo Neue Dateiliste wurde heruntergeladen.
if "%NEUGEHOLT%"=="0" echo Dateiliste ist identisch zum letzten Download.

:: ----Raus bei Fehler -----
if not exist "%SOFTINFOFILEPATH%" goto :ERROR-NOFILE
::

call :WRITE_DATE_OF_FILE "%SOFTINFOFILEPATH%"
echo Downloadliste ist vom %DATE_OF_FILE%
ping 127.0.0.1 -n 4 >nul

::--Kommentarzeilen loeschen -----
type "%SOFTINFOFILEPATH%"| find /V "###" > "%SOFTINFO2-TMP%"
::
::--%20 durch %%20 - wg.Leerzeichen ersetzen-----
"%WORKDIR%\sed" -c 's/%%/%%/g' "%SOFTINFO2-TMP%" > "%SOFTINFO-TMP%"
REM %20 wird sonst als Parameter %2 und 0 interpretiert
::
:: ----alte Infodatei entfernen-fuer email-----
if exist "%WGET-COUNTINFO-TMP%" del "%WGET-COUNTINFO-TMP%"
::
echo.
echo Dateien zum Download in der aktuellen Liste
echo -----
type %SOFTINFO2-TMP%
echo -----
echo.
echo Download und Entpacken der Dateien ...

::-----alte tmp-Log-Datei entfernen---
if exist %LOGNEWFILES-PART% del %LOGNEWFILES-PART%

::-----html-Ausgabe-vorbereiten-----
echo.
echo Letzter Abgleich mit VoBS am %DATE% um %TIME% >> "%SOFTINFO-PART%"
echo.
echo Downloadliste ist vom: %DATE_OF_FILE% ^<br^> >> "%SOFTINFO-PART%"
if "%NEUGEHOLT%"=="0" echo Dateiliste ist identisch und schon vorhandenen.>> "%SOFTINFO-PART%"
if "%NEUGEHOLT%"=="1" echo Dateiliste wurde soeben neu heruntergeladen. >> "%SOFTINFO-PART%"
echo.
echo ^<br^> >> "%SOFTINFO-PART%"
echo ^<h2 class="abstand10px">Aktuelle Programme^</h2^> >> "%SOFTINFO-PART%"
echo ^<ul^> >> "%SOFTINFO-PART%"
ping 127.0.0.1 -n 3 >nul
::-----Programme holen und html-Ausgabe-schreiben-----
set ANZGEHOLT=0
FOR /f %f IN ('type "%SOFTINFO-TMP%') DO (
TITLE autodownload holt Sicherheitsupdates
set DONEFILENAME=UNKNOWN
set DONEFILENAME2=UNKNOWN
rem Dateiname fuer Markerdatei erzeugen
call :GET_DONEFILENAME "%f"
call :GET_MYFILE "%f"
rem schreibt Zeile
call :UNZIP_MYFILE "%f"
ping 127.0.0.1 -n 2 >nul
)
TITLE AUTOMATISIERTER DOWNLOAD fuer Softwareverteilung

::-----html-Ausgabe-schreiben-----
echo ^</ul^> >> "%SOFTINFO-PART%"

::-----Bildschirminfo-----
echo.
echo.
echo =====
echo ---- Download und Entpacken erledigt ----
echo =====
echo.
echo Dateien sind entpackt in Unterordnern von:
echo --^>%SOFTDEPLOYPATH%
echo.

```

```

echo INFOS sind in:
echo --^>"_autodownload\AKTUELLE_PROGRAMME_.html"
echo -----
echo.

:: --- html-Ausgabe-schreiben info-Datei -----
echo Die Programme wurden heruntergeladen und in die entsprechenden Verzeichnisse entpackt. ^<br^>
>> "%SOFTINFO-PART%"
echo Basisverzeichnis: ^"%SOFTDEPLOYPATH%^"
^<br^> >> "%SOFTINFO-PART%"
echo INFO-Dateien in: ^"%SOFTDEPLOYPATH%\_autodownload"^
>> "%SOFTINFO-PART%"
::
echo.
REM -----
REM ---Infodatei der BEREIT-Zeilen erzeugen-und Infomeldung-----
REM -----
echo.-----
call :PREPARE-READY-LIST
echo.-----
::
REM ---Infodatei fertigstellen-----
copy "%WORKDIR%\_html-head.txt" + %SOFTINFO-PART% + "%WORKDIR%\_html-mid.txt" + %READYINFO-PART% +
"%WORKDIR%\_html-foot.txt" "%PROGINFOHTML%" > nul
::
REM -----
REM ---email versenden-----
REM -----
:: ---Neue Updates zaehlen-----
echo 0 > "%TEMP%\tmp-COUNT.txt"
if exist "%WGET-COUNTINFO-TMP%" type "%WGET-COUNTINFO-TMP%" | find /c "WGET-SUCCESS-" > "%TEMP%\tmp-
COUNT.txt"
set /p "ANZ-WGET-GEHOLT="<"%TEMP%\tmp-COUNT.txt"
REM echo "ANZ-WGET-GEHOLT"="%ANZ-WGET-GEHOLT%"
REM pause
::
:: --- Raus bei Markerdatei-1-----
if exist "%WORKDIR%\SEND-NO-EMAIL.txt" (
    echo Datei SEND-NO-EMAIL.txt gefunden: Sende keine email
    echo Datei SEND-NO-EMAIL.txt gefunden: Sende keine email >> "%LOGFILE%"
    goto :DO-NOT-SEND-EMAIL
)
::--- email bei neuen Updates-----
if "%ANZ-WGET-GEHOLT%" NEQ "0 " goto :PREPARE_EMAIL
::
::--- Raus bei Markerdatei-2-----
if exist "%WORKDIR%\SEND-NO-READY-EMAIL.txt" (
    echo Datei SEND-NO-READY-EMAIL.txt gefunden: Sende keine email
    echo Datei SEND-NO-READY-EMAIL.txt gefunden: Sende keine email >> "%LOGFILE%"
    goto :DO-NOT-SEND-EMAIL
)
::--- email bei alten Updates-----
if "%WORKTODO%" EQU "YES" goto :PREPARE_EMAIL
goto :DO-NOT-SEND-EMAIL
::
:PREPARE EMAIL
::--- kein Protokoll ohne neue Updates-----
REM if "%ANZ-WGET-GEHOLT%" EQU "0 " goto :NO_NEW_PROTO
if "%ANZ-WGET-GEHOLT%" EQU "0" goto :NO_NEW_PROTO
REM Datei LOGNEWFILES initialisieren
::-----Protokoll-Ausgabe-loeschen und vorbereiten-----
echo ----- >%LOGNEWFILES%
echo --- Updateprotokoll-[%DATE%]-[%TIME%] >>%LOGNEWFILES%
echo ----- >>%LOGNEWFILES%
::
::---Update-Infos in Protokolldatei schreiben
if exist %LOGNEWFILES-PART% type %LOGNEWFILES-PART% >>%LOGNEWFILES%
if exist %LOGNEWFILES-PART% del %LOGNEWFILES-PART%
:NO_NEW_PROTO
::
::---Meldung wenn PS-Skript fehlt-----
::
if not exist "%PATH-TO-MAIL-PS1%" (
    echo Sende keine email: PS-Skript ist nicht vorhanden
    echo Sende keine email: PS-Skript ist nicht vorhanden >> "%LOGFILE%"
    goto :DO-NOT-SEND-EMAIL
)
::--- Betreff zusammensetzen-----
Set UPATEINFOSTRING='%ANZ-WGET-GEHOLT% Sicherheits-Updates heruntergeladen: Bitte installieren'

```

```

:-- MailBody zusammensetzen -----
Set BODY='Autodownloader - Neue Updates sind bereit - Infos siehe Anhang'

REM === TEST---mit und ohne Leerzeichen je nach Server---???---
if "%ANZ-WGET-GEHOLT%" EQU "1" Set UPATEINFOSTRING='%ANZ-WGET-GEHOLT% Sicherheits-Update
heruntergeladen: Bitte installieren'
rem ===

:: Wenn nur alte BEREIT-Dateien da sind
:-- bei "0" wird "0 " mit Leerzeichen erzeugt - daher beides
if "%ANZ-WGET-GEHOLT%" EQU "0 " Set UPATEINFOSTRING= 'Sicherheits-Update BEREIT: Bitte
kontrollieren'
if "%ANZ-WGET-GEHOLT%" EQU "0" Set UPATEINFOSTRING= 'Sicherheits-Update BEREIT: Bitte kontrollieren'
::
:-- email senden-----

PowerShell.exe -ExecutionPolicy Bypass %PATH-TO-MAIL-PS1% -EmailSubject %UPATEINFOSTRING% -EmailBody
%BODY% -File %PROGINFOHTML%

::
:DO-NOT-SEND-EMAIL

REM ---Aufräumen-----
if exist "%SOFTINFO-TMP%" del "%SOFTINFO-TMP%" > nul
if exist "%SOFTINFO2-TMP%" del "%SOFTINFO2-TMP%" > nul
rem del "%SOFTINFOFILEPATH%" > nul
rem del "%PROGINFO%" > nul
if exist "%SOFTINFO-PART%" del "%SOFTINFO-PART%" > nul
::
REM ---Variable freigeben-----
set SOFTDIR=
set WORKDIR=
set LOGFILE=
set SOFTINFOFILENAME=
set SOFTINFOFILEPATH=
set SOFTINFO-TMP=
set SOFTINFO2-TMP=
set PROGINFO=
set PROGINFOHTML=
set SOFTINFO-PART=
set VOBSPATH=
set VOBSINFOFILEPATH=
set NEUGEHOLT=
set DONEFILENAME=
set SOFTDEPLOYPATH=
set DATE OF FILE=
set ANZGEHOLT=
set WORKTODO=
set READYFILES-TEMP=
set READYINFO-PART=
echo.
popd
echo ---Ende in 15 Sec---
ping 127.0.0.1 -n 15 > nul
rem pause
exit

:--SUB-----
:WRITE_DATE_OF_FILE
rem schreibt Datum der Updatedatei
rem setlocal
set DATE_OF_FILE=%~t1
REM echo Downloadliste ist vom %DATE_OF_FILE%
REM wird direkt geschrieben
::aufräumen
REM set DATE_OF_FILE=
rem endlocal
goto :EOF
:--SUB-END-----

:--SUB-----
:GET_DONEFILENAME
rem zerlegt in Verzeichnis und Dateiname
rem setlocal
set MYFILE=%1
echo.

```

```

echo -----
echo [%MYFILE%]
For /f "delims=/ tokens=1,2" %%i in (%MYFILE%) do (
    Set "MYPREPATH=%%i"
    Set "MYZIPFILE=%%j"
)
rem - Subdir - zb: adobe
rem - Filename - zb: AdobeReader_11.0.03.zip
rem - Filename: letzte 4 Stellen: .zip wegschneiden
rem --
set DONEFILENAME=BEREIT-%%MYPREPATH%-%%MYZIPFILE:~0,-4%.txt
set DONEFILENAME2=ERLEDIGT-%%MYPREPATH%-%%MYZIPFILE:~0,-4%.txt
rem echo %DONEFILENAME%
::aufraeumen
Set MYPREPATH=
Set MYZIPFILE=
rem endlocal
goto :EOF
:--SUB-END-----

:--SUB-----
:GET_MYFILE
setlocal
::
REM ---Schreibe in html-Info-Datei---
set MYFILE=%1
set MYFILE_NOZIP=%%MYFILE:~0,-5%"
rem - wegen html Trick mit " notwendig
echo ^<li^> %MYFILE_NOZIP% ^</li^>    >> "%SOFTINFO-PART%"
::
:--Ausstieg wenn schon da--
set INFOTXT=Datei schon vorhanden [%DONEFILENAME%]
if exist %DONEFILENAME% goto :EXIT_GET_MYFILE
set INFOTXT=Datei schon vorhanden [%DONEFILENAME2%]
if exist %DONEFILENAME2% goto :EXIT_GET_MYFILE
::
echo Hole: %MYFILE%
set MYFILEPATH=%VOBSPATH%/%%MYFILE%
2>"%LOGERROR%" "%WORKDIR%\wget" -a "%LOGFILE%" %MYFILEPATH% -N --progress=dot:binary
set /a FehlerCode=%ERRORLEVEL%
rem echo FehlerCode ist "%FehlerCode%"
if %FehlerCode%==0 goto :WgetSuccess
set INFOTXT=*** Fehler beim Download ***
goto :ErrorWget
::
:WgetSuccess
set INFOTXT=Download erfolgreich
echo WGET-SUCCESS-%%INFOTXT%    >> "%WGET-COUNTINFO-TMP%"
REM Zusatzzeile in html-Datei
echo. ^&nbsp;^; ^&nbsp;^;    [ ^>^>^> NEU heruntergeladen ] >> "%SOFTINFO-PART%"
goto :EXIT_GET_MYFILE
::
:ErrorWget
REM Zusatzzeile in html-Datei
echo. ^&nbsp;^; ^&nbsp;^;    [ ^>^>^> *** Fehler beim Download *** ] >> "%SOFTINFO-PART%"
::
:EXIT_GET_MYFILE
echo %INFOTXT%
echo %INFOTXT% >> "%LOGFILE%"
::
endlocal
goto :EOF
:--SUB-END-----

:--SUB-----
:UNZIP_MYFILE
setlocal
set MYFILE=%1
ping 127.0.0.1 -n 1 >nul
For /f "delims=/ tokens=1,2" %%i in (%MYFILE%) do (
    Set "MYPREPATH=%%i"
    Set "MYZIPFILE=%%j"
)
set UPDFILE=%%MYZIPFILE%
if not exist %UPDFILE% goto :ZIPERROR
REM Info schreiben dass Datei neu ist
echo neues Update: %UPDFILE:~0,-4% >>%LOGNEFILES-PART%
REM Zusatzzeile in html-Datei
REM echo. ^&nbsp;^; ^&nbsp;^;    [ ^>^>^> NEU heruntergeladen ] >> "%SOFTINFO-PART%"
echo entpacke %UPDFILE%

```

```

TITLE Entpacke %MYZIPFILE%
%WORKDIR%\7za t %UPDFILE% * -r >> "%LOGFILE%"
If Errorlevel 1 goto :ZIPERROR
:: exitcode pruefen (7za liefert leider beim List alleine
:: bei nur fehlerhaftem zip keinen Errorlevel
:: Daher diese zusaetzliche Protokollierung.)
::
::--nun in passendes Verzeichnis entpacken und ueberschreiben ---

%WORKDIR%\7za x %UPDFILE% -y -o%SOFTDEPLOYPATH%\%MYPREPATH% >> "%LOGFILE%"
If Errorlevel 1 goto :ZIPERROR

REM TODO_Datei nach AUTODOWNLOAD kopieren
echo kopiere Info "TODO *.pdf" - sofern vorhanden
xcopy "%SOFTDEPLOYPATH%\%MYPREPATH%\%UPDFILE:~0,-4%\TODO*.pdf" /C/Q/H/R/Y
::
REM Nachbearbeitung kopieren
set INFO_POST=Keine Dateien fuer Nachbearbeitung vorhanden und notwendig.
if exist "%SOFTDEPLOYPATH%\%MYPREPATH%\%UPDFILE:~0,-4%\_Nachbearbeitung\." goto DO_COPY_POST
goto WRITE_INFO
:DO_COPY_POST
echo kopiere Daten in "_Nachbearbeitung"
set INFO_POST=Dateien fuer Nachbearbeitung wurden in "_Nachbearbeitung" kopiert.
xcopy "%SOFTDEPLOYPATH%\%MYPREPATH%\%UPDFILE:~0,-4%\_Nachbearbeitung\*.*"
"%SOFTDEPLOYPATH%\_NACHBEARBEITUNG\*.*" /K/R/E/I/S/C/H/Y/Q

:-----
:WRITE_INFO
:--wenn ok--
echo ----- >> %DONEFILENAME%
echo automatisch generiert von autosoftload %VER% >> %DONEFILENAME%
echo ----- >> %DONEFILENAME%
echo Datei "%MYZIPFILE%" wurde vom VoBS geholt und entpackt >> %DONEFILENAME%
echo - nach: %SOFTDEPLOYPATH%\%MYPREPATH%\%UPDFILE:~0,-4% >> %DONEFILENAME%
echo - am : %date% um %time% >> %DONEFILENAME%
echo - %INFO_POST% >> %DONEFILENAME%
echo ----- >> %DONEFILENAME%
echo HINWEISE: >> %DONEFILENAME%
echo -1- Aktueller Stand siehe: "_AKTUELLE_PROGRAMME_.html" >> %DONEFILENAME%
echo -2- Die Textdatei "%DONEFILENAME%" >> %DONEFILENAME%
echo verhindert den erneuten Download des Updates vom VoBS >> %DONEFILENAME%
echo -3- Update bitte umgehend in die Softwareverteilung integrieren! >> %DONEFILENAME%
echo -4- optional: Nach Integration umbenennen von BEREIT-... zu ERLEDIGT-... >> %DONEFILENAME%
echo Die umbenannte Textdatei "%DONEFILENAME2%" >> %DONEFILENAME%
echo verhindert ebenso den erneuten Download des Updates >> %DONEFILENAME%
echo -5- Ersetzte Updates in Softwareverteilung entfernen und >> %DONEFILENAME%
echo dann ersetze Updates bitte aus Platzgruenden loeschen! >> %DONEFILENAME%
echo -6- Textdatei [siehe -2- und -4-] soll/kann geloescht werden >> %DONEFILENAME%
echo wenn das Nachfolgeupdate da ist. >> %DONEFILENAME%
echo -7- Nachfolgeupdate: gleicher Name andere Versionsnummer >> %DONEFILENAME%
echo ----- >> %DONEFILENAME%
:--Platz sparen - zip-Datei entfernen
echo Saubermachen: Entferne "%MYZIPFILE%"
del %UPDFILE%
::
:--bei Fehler---
:ZIPERROR
endlocal
goto :EOF
:--SUB-END-----

:--SUB-----
:PREPARE-READY-LIST
rem erzeugt html-Liste der BEREIT-Dateien
rem setlocal
REM set READYFILES-TEMP=%TEMP%\tmp-DirReadyFiles.txt
REM set READYINFO-PART=%TEMP%\READYINFO-PART.txt
if exist "%SOFTDIR%\BEREIT-*.txt" goto :GETREADYINFO
set WORKTODO=NO
:--Lobmeldung-weil-keine BEREIT-Datei-existiert----
echo ^<li^> >> "%READYINFO-PART%"
echo Bestens - alles erledigt: Keine Sicherheitsupdates mit Status BEREIT gefunden >
"%READYINFO-PART%"
echo Bestens - alles erledigt: Keine Sicherheitsupdates mit Status BEREIT gefunden >>
"%LOGFILE%"
echo Bestens - alles erledigt: Keine Sicherheitsupdates mit Status BEREIT gefunden
echo ^</li^> >> "%READYINFO-PART%"
goto :PREPAREREADYDONE
::
:GETREADYINFO

```



```

:-- Liste mit BEREIT-Dateien erzeugen--
:--Initialisieren
set WORKTODO=YES
if exist "%READYINFO-PART%" del "%READYINFO-PART%"
dir /b "%SOFTDIR%\BEREIT-*.txt" > %READYFILES-TEMP%
REM -----html Datei erzeugen-----
For /f %%a in (%READYFILES-TEMP%) do (
    echo ^<li^>    >> "%READYINFO-PART%"
    echo %%~na    >> "%READYINFO-PART%"
    echo %%~na
    echo * %%~na >> "%LOGFILE%"
    echo ^</li^> >> "%READYINFO-PART%"
)
::aufraeumen
:PREPAREREADYDONE
Set MYPREPATH=
Set MYZIPFILE=
if exist "%READYFILES-TEMP%" del "%READYFILES-TEMP%"
rem endlocal
goto :EOF
:--SUB-END-----

:-----
:ERROR-NOFILE
echo.
echo ---Problem-----
echo.
echo Es konnte keine Liste vom VoBS geholt werden!
echo.
echo -----
echo.                                     >> "%LOGFILE%"
echo ----- >> "%LOGFILE%"
echo FEHLER: Es konnte keine Liste vom VoBS geholt werden! >> "%LOGFILE%"
echo ----- >> "%LOGFILE%"
echo.

:GANZ_ENDE
popd
echo ---Ende in 60 Sec---
ping 127.0.0.1 -n 60 > nul

```

3.2.2. Host1

3.2.2.1. [EMail-Alert HD-Fehler.ps1](#)

```

#Dieses Skript funktioniert nur zusammen mit dem Email-Skript PSMail30

#Variable für Mail-Betreff
$DatumZeit = Get-Date
$Sender = $env:COMPUTERNAME
$Betreff = "Event HD-defekt (Smart Array Event 24595)"

#Email Body zusammenfügen
$Body0 = "Dieses Mail wurde vom $Sender gesendet."
$Body1 = "Ereigniszeitpunkt: $DatumZeit"
$Body2 = " "
$Body3 = "Smart Array vom $Sender hat ein Problem festgestellt."
$Body4 = " "
$Body5 = "Versandt mit PSMail30.ps1"

$Body = "$Body0<br>$Body1<br>$Body2<br>$Body3<br>$Body4<br><br>$Body5"

#Email senden
Set-Location C:\_Setup\RB-PSMail\RB-PSMail
.\PSMail30.ps1 -EmailSubject "$Betreff" -EmailBody $Body

```

3.2.2.2. [EMail-Alert HD-Fehler1216.ps1](#)

```
#Dieses Skript funktioniert nur zusammen mit dem Email-Skript PSMail30

#Variable für Mail-Betreff
$DatumZeit = Get-Date
$Sender = $env:COMPUTERNAME
$Betreff = "Event HD-defekt (Fehler ID 1216)"

#Email Body zusammenfügen
$Body0 = "Dieses Mail wurde vom $Sender gesendet."
$Body1 = "Ereigniszeitpunkt: $DatumZeit"
$Body2 = " "
$Body3 = "Storage Agents vom $Sender hat ein Problem festgestellt."
$Body4 = " "
$Body5 = "Versandt mit PSMail30.ps1"

$Body = "$Body0<br>$Body1<br>$Body2<br>$Body3<br>$Body4<br><br>$Body5"

#Email senden
Set-Location C:\_Setup\RB-PSMail\RB-PSMail
.\PSMail30.ps1 -EmailSubject "$Betreff" -EmailBody $Body
```

3.2.2.3. [EMail-Alert HD-Fehler4879.ps1](#)

```
#Dieses Skript funktioniert nur zusammen mit dem Email-Skript PSMail30

#Variable für Mail-Betreff
$DatumZeit = Get-Date
$Sender = $env:COMPUTERNAME
$Betreff = "Event HD-defekt (Fehler ID 4879)"

#Email Body zusammenfügen
$Body0 = "Dieses Mail wurde vom $Sender gesendet."
$Body1 = "Ereigniszeitpunkt: $DatumZeit"
$Body2 = " "
$Body3 = "Agentless Mangement Service vom $Sender hat ein Problem festgestellt."
$Body4 = " "
$Body5 = "Versandt mit PSMail30.ps1"

$Body = "$Body0<br>$Body1<br>$Body2<br>$Body3<br>$Body4<br><br>$Body5"

#Email senden
Set-Location C:\_Setup\RB-PSMail\RB-PSMail
.\PSMail30.ps1 -EmailSubject "$Betreff" -EmailBody $Body
```

3.2.2.4. [EMail-Alert HD-FehlerWarnung.ps1](#)

```
#Dieses Skript funktioniert nur zusammen mit dem Email-Skript PSMail30

#Variable für Mail-Betreff
$DatumZeit = Get-Date
$Sender = $env:COMPUTERNAME
$Betreff = "Event HD-defekt (Fehler ID 1200)"

#Email Body zusammenfügen
$Body0 = "Dieses Mail wurde vom $Sender gesendet."
$Body1 = "Ereigniszeitpunkt: $DatumZeit"
$Body2 = " "
$Body3 = "Storage Agents vom $Sender hat ein Problem festgestellt."
$Body4 = " "
$Body5 = "Versandt mit PSMail30.ps1"

$Body = "$Body0<br>$Body1<br>$Body2<br>$Body3<br>$Body4<br><br>$Body5"

#Email senden
Set-Location C:\_Setup\RB-PSMail\RB-PSMail
.\PSMail30.ps1 -EmailSubject "$Betreff" -EmailBody $Body
```

3.2.2.5. [EMail-Alert Replikationsfehler \(ein Skript für alle Fehler\)](#)

```
#Dieses Skript funktioniert nur zusammen mit dem Email-Skript PSMail30

#Variable für Mail-Betreff
$DatumZeit = Get-Date
$Sender = $env:COMPUTERNAME
$Betreff = "Event Replikationsfehler"

#Email Body zusammenfügen
$Body0 = "Dieses Mail wurde vom $Sender gesendet."
$Body1 = "Ereigniszeitpunkt: $DatumZeit"
$Body2 = " "
$Body3 = "System vom $Sender hat ein Problem festgestellt."
$Body4 = "Fehler-ID: 10028, 29292, 32022, 32086, 32088, 32315, 32570 oder 33680"
$Body5 = "Versandt mit PSMail30.ps1"

$Body = "$Body0<br>$Body1<br>$Body2<br>$Body3<br>$Body4<br><br>$Body5"

#Email senden
Set-Location C:\_Setup\RB-PSMail\RB-PSMail
.\PSMail30.ps1 -EmailSubject "$Betreff" -EmailBody $Body
```

3.2.2.6. RB-PSMail ServerStatusRep.ps1

```
#Dieses Skript funktioniert nur zusammen mit dem Email-Skript PSMail30

#Variable
#Namen der virtuellen Maschinen eintragen
$Datum = Get-Date
$Sender = $env:COMPUTERNAME
$DC = "DCSchule"
$SRV = "Server"

#VM-Status und Replikationsstatus
$StatusDC = (Get-VM -Name $DC).State
$StatusSRV = (Get-VM -Name $SRV).State

$ReplicationHealth = (Get-VMReplication -VMNAME $SRV)
if ($ReplicationHealth.Health -eq "Normal") {
    $HealthSRV = "Normal"
}
else {
    $HealthSRV = "Kritisch oder Warnung!"
}

$ReplicationHealth = (Get-VMReplication -VMNAME $DC)
if ($ReplicationHealth.Health -eq "Normal") {
    $HealthDC = "Normal"
}
else {
    $HealthDC = "Kritisch oder Warnung!"
}

$Body1b = "VM $SRV <b>$StatusSRV</b><br>VM $DC <b>$StatusDC</b><br>"
$Body1 = "Replikationsstatus $SRV <b>$HealthSRV</b><br>Replikationsstatus $DC <b>$HealthDC</b><br>"

# Freier Speicher auf Server, DC und host1 auslesen
$SRVC1 = Get-WmiObject win32_logicaldisk -Computer Server -filter "name='c:'" | select name,
volumename, freespace
$SRVC2 = $SRVC1.freespace
$SRVC = [math]::Round($SRVC2/1GB,2)

$SRVD1 = Get-WmiObject win32_logicaldisk -Computer Server -filter "name='d:'" | select name,
volumename, freespace
$SRVD2 = $SRVD1.freespace
$SRVD = [math]::Round($SRVD2/1GB,2)

$DCC1 = Get-WmiObject win32_logicaldisk -Computer DCSchule -filter "name='c:'" | select name,
volumename, freespace
$DCC2 = $DCC1.freespace
$DCC = [math]::Round($DCC2/1GB,2)

$HOSTC1 = Get-WmiObject win32_logicaldisk -filter "name='c:'" | select name, volumename, freespace
$HOSTC2 = $HOSTC1.freespace
$HOSTC = [math]::Round($HOSTC2/1GB,2)

$HOSTD1 = Get-WmiObject win32_logicaldisk -filter "name='d:'" | select name, volumename, freespace
$HOSTD2 = $HOSTD1.freespace
$HOSTD = [math]::Round($HOSTD2/1GB,2)

$Body2 = "Freier Speicher auf... <br>Server C: $SRVC GB <br>Server D: $SRVD GB <br>DCSchule C: $DCC
GB <br>HOST1 C: $HOSTC GB <br>HOST1 D: $HOSTD GB <br>"

#WSUS läuft
$wsus = get-service *wsus* -computername server
if($wsus.status -eq "Running")
    {$WSUSstatus="WSUS läuft!"}
else
    {$WSUSstatus="WSUS läuft NICHT!!"}

#WDS läuft
$wds = get-service WDSserver -computername server
if($wds.status -eq "Running")
    {$WDSstatus="WDS läuft!"}
else
    {$WDSstatus="WDS läuft NICHT!!"}

#Body für Dienste-Status zusammensetzen
$Body3 = "$WSUSstatus<br>$WDSstatus"

#WSUS Größe Datenbank und Contentordner
$WSUSCont = (Get-ChildItem -Path \\server\CDS\WSUS\wsusContent -Force -Recurse -ErrorAction
silentlyContinue | Measure-Object -Property Length -Sum).Sum
$WSUSDB = (get-childitem \\server\c$\Windows\WID\Data\SUSDB.mdf | select-object length).length

$Body4 = 'WSUS Content Ordnergröße: {0:n2} GB' -f ($WSUSCont/1GB)
$Body5 = 'WSUS Datenbankgröße SUSDB.mdf: {0:n2} GB' -f ($WSUSDB/1GB)

#Softwareupdates bereit
$SWUP = ((Get-ChildItem \\server\Softwareverteilung\_autodownload\BEREIT*.*)).Name -join "<br>"
if ($SWUP -eq $null)
    {$Body6 = "KEINE Softwareupdates BEREIT!"}
else
```

```
{ $Body6 = "Softwareupdates BEREIT:<br>$SWUP" }
```

```
#Email Body zusammenfügen
```

```
$Body0 = "Dieses Mail wurde vom $Sender gesendet <br>... $Datum<br>"
```

```
$Body = "$Body0<br>$Body1b<br>$Body1<br>$Body2<br>$Body3<br><br>$Body4<br>$Body5<br><br>$Body6"
```

```
#Email senden
```

```
Push-Location C:\_Setup\RB-PSMail\RB-PSMail
```

```
.\PSMail30.ps1 -EmailSubject "Server Statusreport - $Datum" -EmailBody $Body
```