MATRIX42
SMARTER WORKSPACE – BETTER LIFE

Silverback by Matrix42

Exchange PowerShell Integration Guide

Version 2.0.0

22. July 2016

Copyright © 2000 - 2014 Matrix42 AG

This documentation is copyright protected. All rights are reserved by Matrix42 AG.

Any other use, in particular the disclosure to third parties, storage in a data system, dissemination, processing, presentation, performance and demonstration are prohibited. This applies to the entire document, as well as parts thereof.

Subject to change. Reprint, also in excerpts, is permitted only with the written consent of Matrix42 AG.

The software described in this document is subject to a permanent development due to which there may be differences in the documentation and the actual software. This documentation is not entitled to the actual functionality of the software.

Apple and **Mac OS X** are registered trademarks of Apple Inc.
Citrix® software or **Citrix® server** are Trademarks and Registered Trademarks of Citrix Systems, Inc. in the United States and other countries.
cygwin is copyrighted by Red Hat Inc. 1996-2003.
expat is copyrighted by Thai Open Source Software Center Ltd.
gSOAP is copyrighted by Robert A. van Engelen, Genivia, Inc. All rights reserved.
Iconv is copyrighted by 1999-2003 Free Software Foundation, Inc.
Iperf is copyrighted by the University of Illinois, except for the gnu_getopt.c, gnu_getopt_long.c, gnu_getopt.h files, and inet_aton.c, which are under the GNU General Public License.
Libmspack (C) 2003-2004 by Stuart Caie <kyzer@4u.net>.
OpenSSL This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit.
PuTTY is copyrighted by Simon Tatham. Portions copyright Robert de Bath, Joris van Rantwijk, Delian Delchev, Andreas Schultz, Jeroen Massar, Wez Furlong, Nicolas Barry, Justin Bradford, Ben Harris, Malcolm Smith, Ahmad Khalifa, Markus Kuhn, and CORE SDI S.A.
RSA Data Security, Inc. MD5 Message-Digest Algorithm is copyrighted by RSA Data Security Inc. Created 1991. All rights reserved.
rsync is an open source utility that provides fast incremental file transfer. rsync is freely available under the GNU General Public License version 2.
runcontrol The Initial Developer of the Original Code is James Clark. Portions created by James Clark are Copyright (c) 1998 James Clark. All rights reserved.
SNMP++ Copyright (c) 1996 Hewlett-Packard Company.
VMware, the **VMware "boxes" logo and design, Virtual SMP, VMotion vSphere, vSphere Hypervisor (ESXi), ESX, View, ThinApp, vCenter** and **vCloud** are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions.
Windows, Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, **Windows Server 2012,** and **Microsoft Exchange** are registered trademarks of Microsoft Corporation.

Others, at this point not explicitly listed, company, brand and product names are trademarks or registered trademarks of their respective owners and are subject to trademark protection.

Inhaltsverzeichnis

1. Introduction 4
2. Architecture 5
   2.1. Silverback cloud based architectural overview 6
   2.2. Silverback on-premise architectural overview 7
   2.3. Silverback hybrid cloud architectural overview 8
3. Prerequisites 9
   3.1. Account requirements and Permissions 9
   3.2. Network And System Requirements 9
   3.3. Recommended Reading 9
4. Configuration 10
   4.1. Enabling PowerShell for Silverback server 10
   4.2. Setting up the Service Account 11
      4.2.1. Create Service Account 11
      4.2.2. Assign Service Account Permissions 12
   4.3. Create the Exchange ActiveSync Mailbox Policy 13
      4.3.1. For Exchange 2010 13
      4.3.2. For Exchange 2013 13
   4.4. Enabling PowerShell for Exchange server 14
   4.5. Testing PowerShell connectivity 15
   4.6. Configuring Silverback PowerShell Settings 16
   4.7. Enable Exchange Protection in the Silverback service 17
      4.7.1. Troubleshooting 18
   4.8. Testing Exchange Quarantine Rules 19
   4.9. Enable Quarantine Rules 20
5. Appendix 21

1. Introduction

This guide will help you Integrate Silverback into a Microsoft Exchange environment using PowerShell. This allows you to control who can access Exchange ActiveSync from Smartphones and Tablets. Silverback acts as the management platform providing control over Exchange ActiveSync access based on its policies. Silverback with either Allow managed devices, or Deny unmanaged devices, ensuring a base level of compliance. This is the simplest Exchange ActiveSync control mechanism, and the quickest to implement. This architecture does not require additional Application Delivery Controllers in your corporate perimeter.

2. Architecture

The Silverback quarantine model integrates into your Exchange environment using Microsoft PowerShell. PowerShell is a configuration management framework from Microsoft built on the .NET Framework, it provides a management layer that enables Silverback to perform administrative tasks on your corporate Exchange environments. Thus allowing Silverback to make policy decision on which devices can access corporate email and which devices cannot.

This model provides a level of control over ActiveSync access and ensures compliance with corporate policy and malicious attackers are prevented from accessing sensitive email information.

To provide this level of protection Silverback takes on an Exchange Administrator roll by issuing PowerShell commands to the Exchange infrastructure, to orchestrate your mobile fleet.

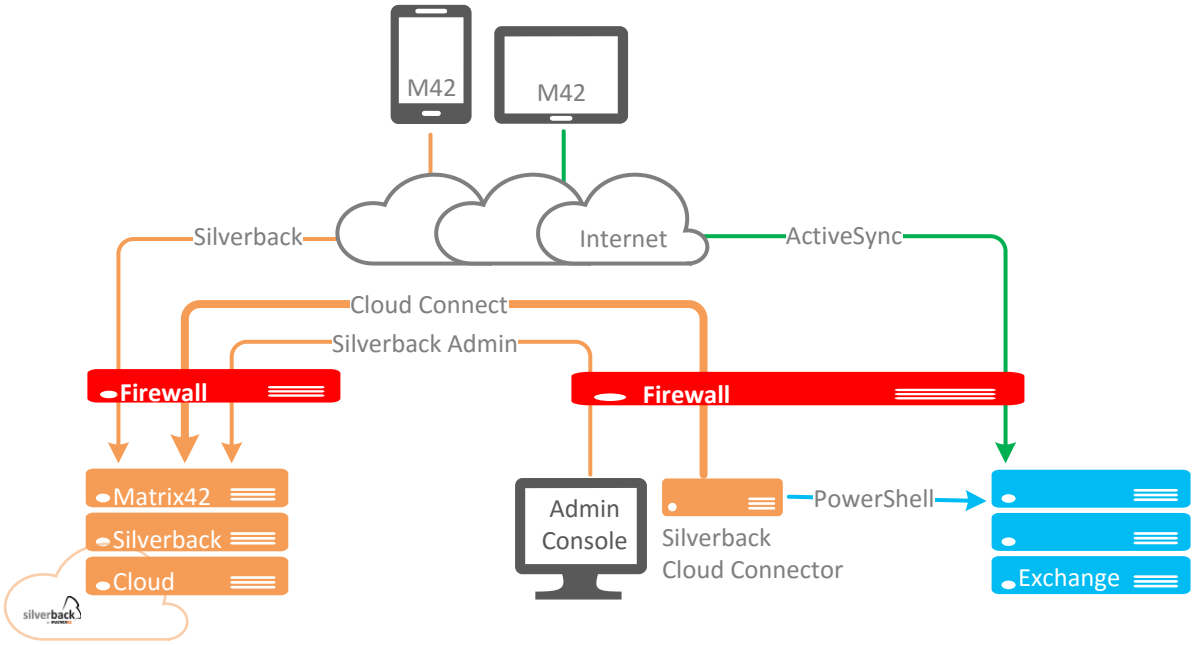
This model allows you to integrate with an on-premise Exchange, cloud Exchange and Hybrid. Here are the supported models:

Matrix 42 Silverback	Exchange ActiveSync	Supported
On-Premise	On-Premise	Yes
Matrix 42 Cloud	On-Premise	Yes
On-Premise	Cloud	Yes
Matrix 42 Cloud	Cloud	Yes

The Silverback Quarantine model utilises PowerShell; therefore the Exchange environment needs to support PowerShell and Quarantining. This means it will not work with Exchange 2003, Lotus Domino or none Exchange based email services.

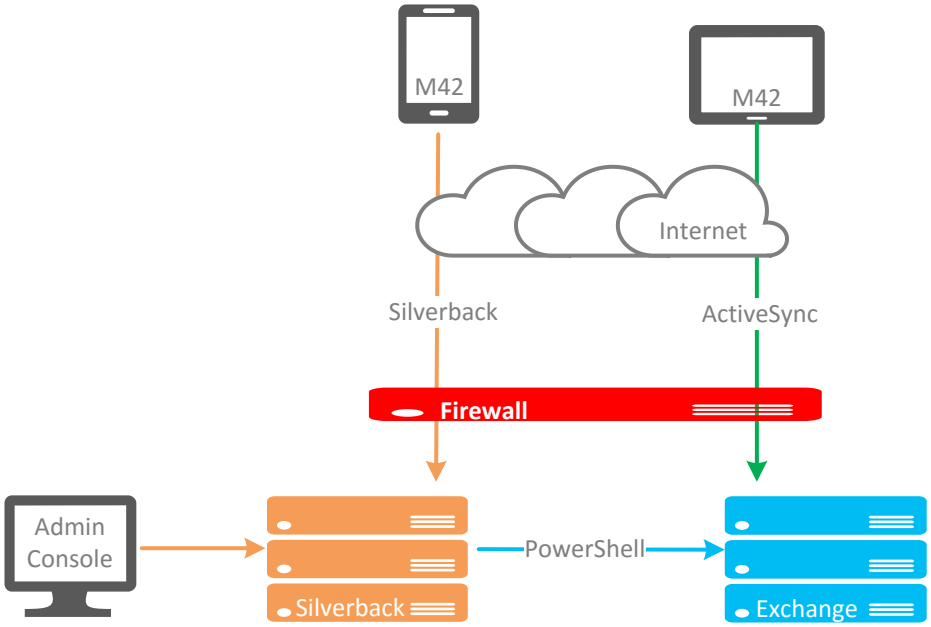
2.1. Silverback cloud based architectural overview

This high-level architecture covers a Matrix 42 cloud based Silverback deployment controlling an on-premise Exchange environment, using the Silverback Cloud Connector.



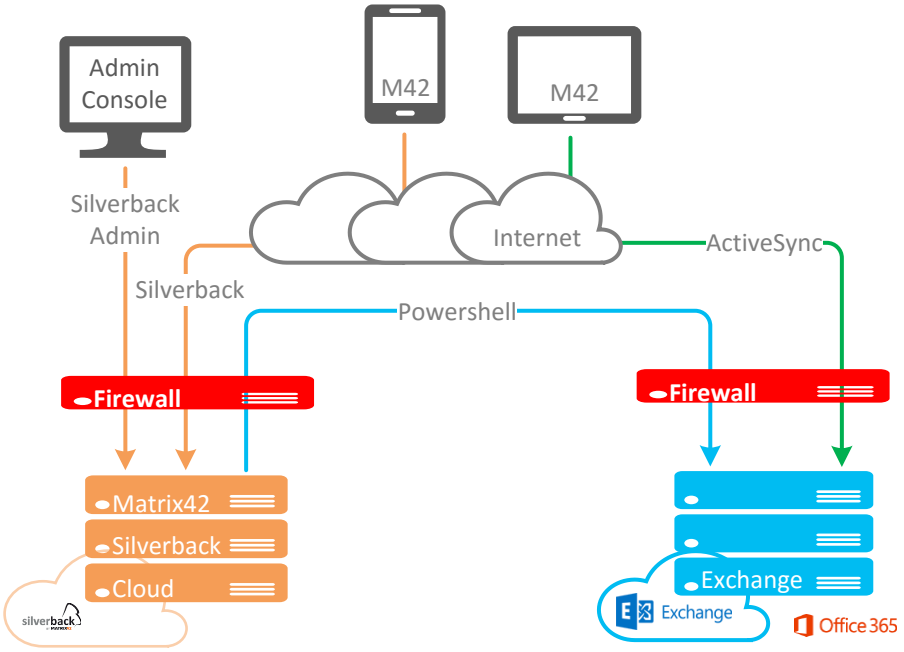
2.2. Silverback on-premise architectural overview

This high Level diagram covers an on-Premise Silverback deployment controlling an On-Promise Exchange environment:



2.3. Silverback hybrid cloud architectural overview

This high Level diagram covers an on-Premise Silverback deployment controlling an On-Promise Exchange environment:



3. Prerequisites

You need to make sure you have the correct account permission and systems requirements before you start setting up PowerShell integration.

3.1. Account requirements and Permissions

Silverback must be assigned the correct permissions to manage the Exchange ActiveSync Devices. Silverback uses a Service Account to manage this. Please do the following steps:

- Create a Service Account in the AD infrastructure of the Target Exchange environment.
- Assign the Service Account the following permissions:
  - Organisation Management
  - Server Management
  - Recipient Management
- Create an Exchange ActiveSync Client Access profile.
  (This will be used to assign to managed devices)

Please store these credentials safely. You will need them in the future.

3.2. Network And System Requirements

Silverback controls exchange through the PowerShell web services running on the Exchange server. Make sure this is configured correctly and running.

3.3. Recommended Reading

If you are an experienced Silverback Administrator you can skip this section, but you should read the latest Silverback Documentation:

- Silverback Admin User Guide
- Silverback Cloud Connector Overview
- Silverback Release Notes

4. Configuration

This will guide you through configuring Exchange, PowerShell and Silverback to successfully manage devices.

4.1. Enabling PowerShell for Silverback server

Windows PowerShell 3.0 arrives, installed, configured and ready to use on Windows Server 2012. PowerShell is bundled as part of Microsoft .Net Framework 4.5. This is a pre-requisite for a Silverback installation. So PowerShell is already installed on the Silverback server. PowerShell is installed in a locked-down mode, and set to restricted execution. You will need to open this up, by setting the Execution Policy to Remote Signed. The PowerShell Integrated Scripting Environment is installed on Windows 2012, but you need to install the Window Management Framework to enable it on Windows 2008. On the Silverback server run the following:

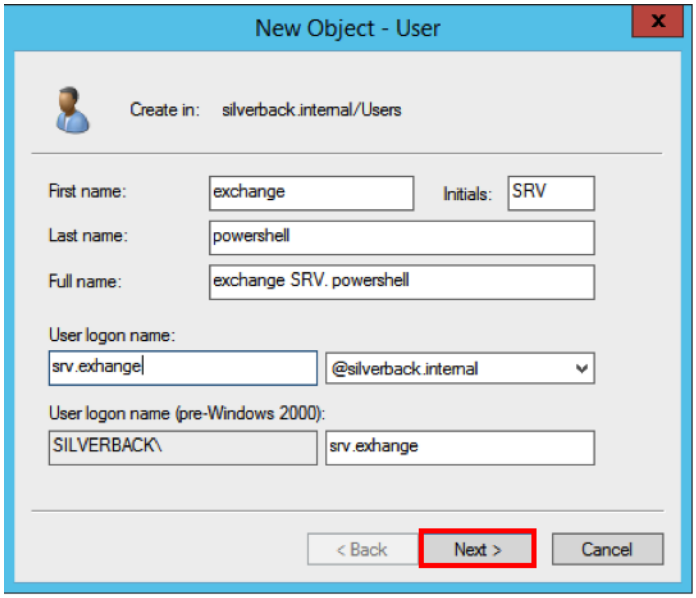
```
PS >
PS > Set-ExecutionPolicy RemoteSigned
PS >
```

4.2. Setting up the Service Account

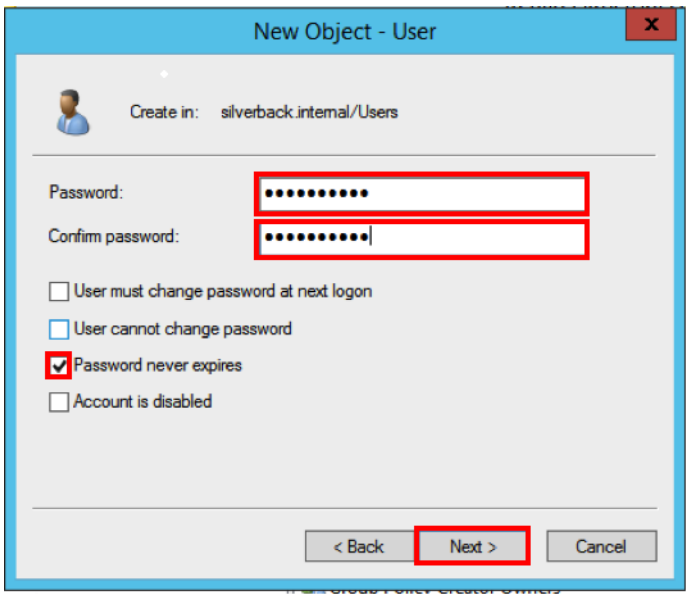
Create a service Account in Active Directory account and assign the appropriate permissions. Open Active Directory User and computer and create a new user:

4.2.1. Create Service Account

Active Directory User and Computers > **New** > **User**. Complete the new user form, click **Next**

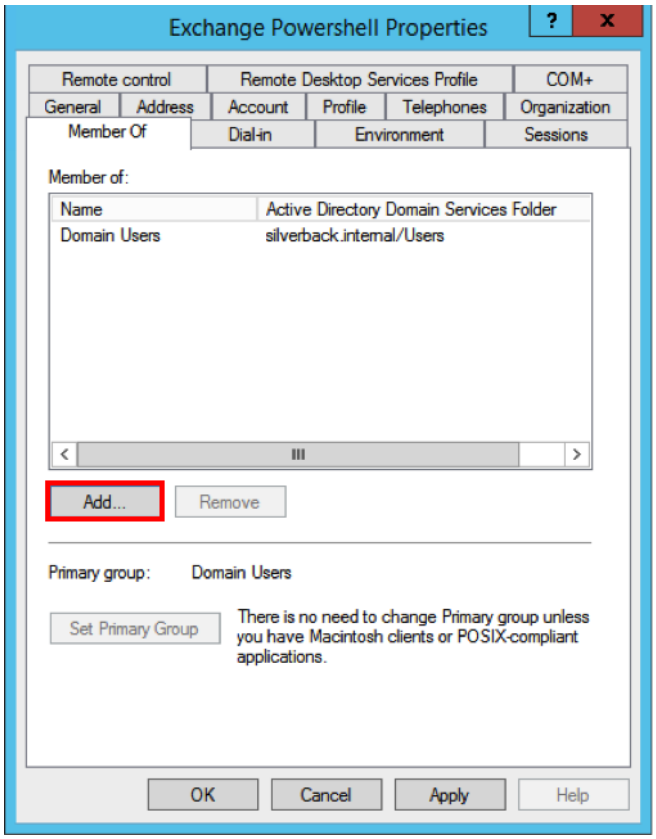


Set a complex password. You will need this later on. Enable “Password never expires”. Click Next

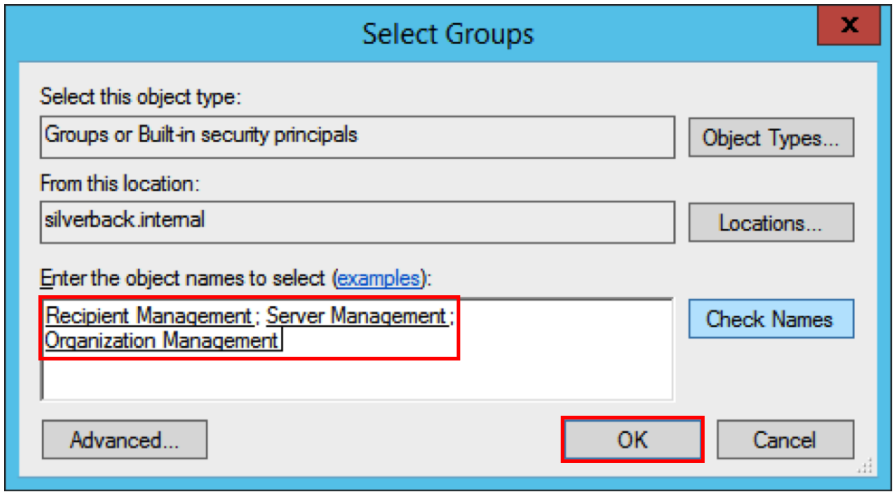


4.2.2. Assign Service Account Permissions

Once the user has been created, edit the users. Navigate to the **'Member Of'** tab and click Add:



- Assign the Service Account the following permissions:
  - Organisation Management
  - Server Management
  - Recipient Management



4.3. Create the Exchange ActiveSync Mailbox Policy

Silverback needs to integrate with Exchange via a client access or mailbox policy. This is assigned to the devices in the system when they enrol. Exchange ActiveSync will have a default policy, but it is best practise to separate this out into a new policy, so it's clear which policy Silverback will interact with.

The process for this is slightly difference depending on your version of Exchange.

4.3.1. For Exchange 2010

1 In the Exchange Management Console, navigate to Organization Configuration and then Client Access
2 Click the Mailbox Polices Tab and in the action pane, click New Mailbox Policy
3 Enter a name for the profile (and record this name, it's required to configure Silverback later).
4 Save the profile.

4.3.2. For Exchange 2013

1 In the Exchange Admin Center, click on Mobile from the left menu, then click on mobile device mailbox policies
2 Click the plus button to add a new policy
3 Enter a name for the profile (and record this name, it's required to configure Silverback later).
4 Save the profile

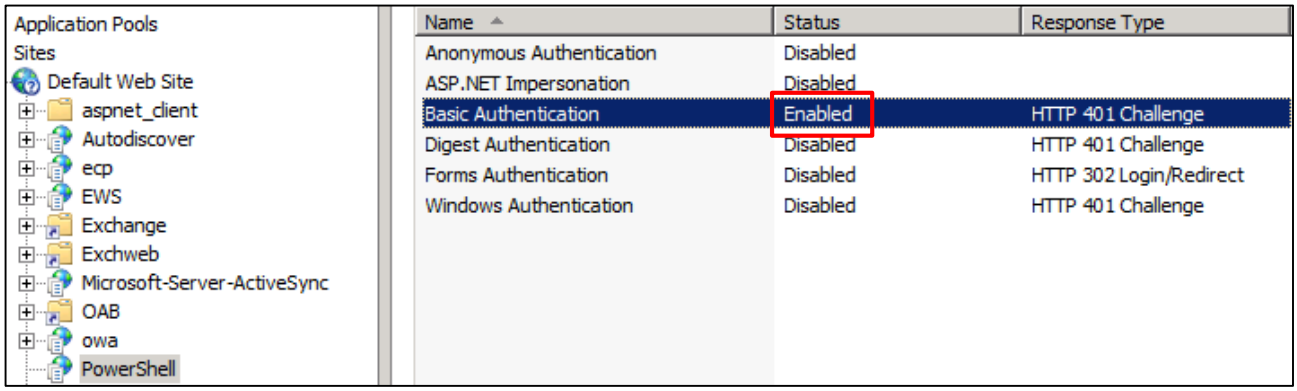
4.4. Enabling PowerShell for Exchange server

Microsoft Exchange has been tightly integrated with Windows PowerShell, you should not have to install anything on the Exchange server for PowerShell to work. The default PowerShell environment will be locked-down so you need to set the Execution Policy to Remote Signed to be allowed to run scripts.

You only need to configure this setting once on your computer, not every time you connect. To enable Windows PowerShell to run signed scripts, run the following command in an elevated Windows PowerShell window (a Windows PowerShell window you opened by selecting Run as administrator).

```
PS > Set-ExecutionPolicy RemoteSigned
```

You should confirm that the PowerShell IIS Application Pool 'PowerShell' has a correctly installed SSL certificate, is started and has 'Basic Authentication' enabled.



4.5. Testing PowerShell connectivity

Once Silverback and Exchange have PowerShell setup correctly it is a good idea to test connectivity. On the Silverback server run the following:

```
PS >
PS > $UserCredentials = Get-Credential
PS >
```

In the Windows PowerShell Credential Request dialog box, type your service account name and password, and then click OK. Now Run:

```
PS >
PS > $session = New-PSSession -ConfigurationName
Microsoft.Exchange -ConnectionUri https://Exchange.Server
/powershell/ -Authentication Basic -Credential $UserCredentials
PS >
```

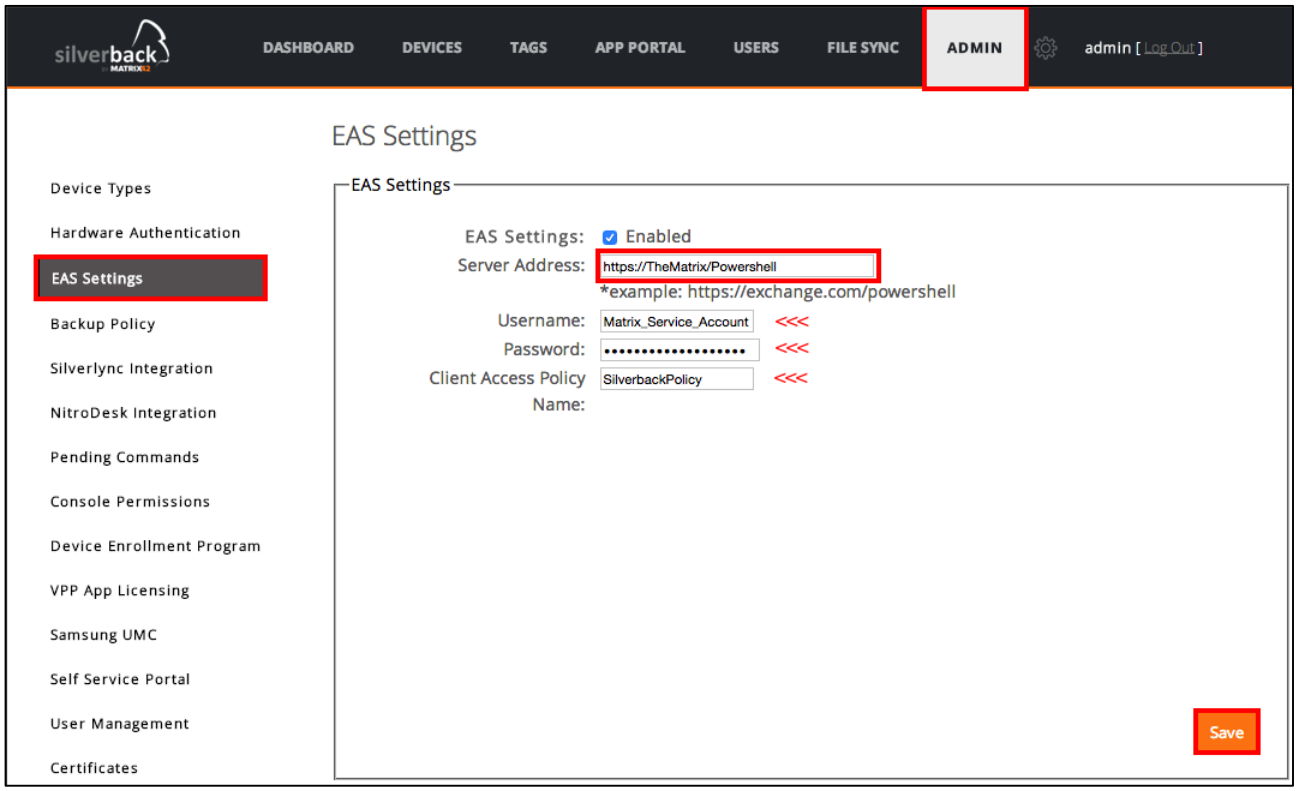
This will build a session to the Exchange Client Access Server: Now Run:

```
PS >
PS > Import-PSSession $Session
PS >
```

You should now be able to run standard Exchange commands like Get-Mailbox

4.6. Configuring Silverback PowerShell Settings

- Log into the Silverback Admin Console
- Navigate to the Admin > EAS Settings section.
- Enable the EAS Settings, and fill out the details.
  - Please add you Exchange server's PowerShell URL
  - Add the Service Account with appropriate privileges
  - Enter the password
  - Specify the Client Access Policy setup to manage devices.



4.7. Enable Exchange Protection in the Silverback service

After configuring Silverback and Exchange, you need to enable the functionality in the Silverback Service. This is performed in the same manner for Cloud Connector and On Premise deployments.

1 Log into the Silverback admin console as a user with the Settings Administrator role
2 Click on the Cloud Connector Settings section
3 Change “Enable Exchange Protection” to “Yes”
4 Set the Exchange Task Interval value to something frequent, e.g. 1 minute (this is the amount of time before Silverback will enable and disable users in quarantine for exchange. The shorter the value, the quicker users will be allowed access and blocked from Exchange ActiveSync).
5 Click Save
6 Restart the Silverback Maintenance Service on the Silverback server

4.7.1. Troubleshooting

Check the following requirements:

- A common problem is an incorrect password. Confirm the credentials.
- The account you use to connect to the Exchange server must be enabled for remote Shell. For more information, see Manage Exchange Management Shell access.
- TCP port 443 needs to be open between your local computer and the Exchange server. It's probably open, but it’s something to consider if your organization has a restrictive network policy. Check your firewall rules.
- If you get an Access Denied error in the logs, but you’ve verified the credentials and address are correct, try configuring the system to use HTTP instead of HTTPS. Some SSL configurations can prevent access.

4.8. Testing Exchange Quarantine Rules

Before you enable quarantining across your Exchange Infrastructure, you should test that the ActiveSync Device IDs are being properly populated. If you do not confirm this, prior to turning it on, you may accidentally block some device from accessing Exchange ActiveSync.

Now that Silverback is configured with the PowerShell URL and service account configured, you need to test Silverback quarantine rules. Enrol a device into Silverback. Please reference the Silverback End User Guide for help with this.

Once a smartphone is enrolled, Silverback will populate its device id into the users mailbox settings. The following PowerShell outputs the device ids that have been granted access to a user's mailbox.

```
PS >
PS > get-casmailbox -Identity "user@domain.com" | select
{$_.ActiveSyncAllowedDeviceIDs}
PS >
```

This should output the devices id’s of all devices enrolled into Silverback.

```
PS >
$_.ActiveSyncAllowedDeviceIDs
-----------------------------
Appl DYVJ26T4DVGK
Appl WAYNE84JYHOE

PS >
```

Silverback is correctly allocating the device ID's, so we need to test an unmanaged device gets blocked. Try to manually setup an ActiveSync device for this user without enrolling them into Silverback. This should not work.

If the device cannot be manually enrolled into Silverback, you have nearly completed the setup. All you need to do now is enable organisational quarantine.

4.9. Enable Quarantine Rules

Silverback enables you to restrict access to Microsoft Exchange ActiveSync by using its quarantine rules. This feature prevents users from synchronizing unauthorized mobile phones with Exchange.

By default, if Exchange ActiveSync is enabled for a user, that user can synchronize the Exchange mailbox with any smartphone. To restrict users to specific smartphones, Silverback populates the ActiveSyncAllowedDeviceIDs parameter from each user. This ensures that only these devices can connect.

NOTE: If you are enabling this on a pre-existing production ActiveSync fleet, please make sure that all your devices are enrolled into Silverback. Ensure all the devices have a device id set in users' mailbox settings. If a user does not have this set they will not be able to connect to ActiveSync.

Once you are sure the entire device fleet is enrolled into Silverback and it's been configured correctly, you can enable Organisational Quarantine.

Just in case you need to rollback these settings, first determine the default access level:

```
PS >
PS > Get-ActiveSyncOrganizationSettings | ft DefaultAccessLevel
PS >
```

The default setting will output:

```

DefaultAccessLevel
------------------
Allow
```

To enable quarantine, open the Windows PowerShell command prompt on the Exchange Server and enter the following command:

```
PS >
PS > Set-ActiveSyncOrganizationSettings – DefaultAccessLevel
quarantine
PS >
```

If you need to revert the settings enter the following command:

```
PS >
PS > Set-ActiveSyncOrganizationSettings – DefaultAccessLevel Allow
PS >
```

5. Appendix

Just a quick post on useful PowerShell commands to get some ActiveSync config and statistics. These are general commands related to ActiveSync Devices

Get device statistics:

```
PS >
PS > Get-MailboxStatistics -Identity username | fl
PS >
```

Get ActiveSync statistics:

```
PS >
PS > Get-ActiveSyncDeviceStatistics -Mailbox username
PS >
```

Get more info for specific user, include ActiveSyncAllowedDeviceIDs configured:

```
PS >
PS > Get-CASMailbox -Identity username | fl *ActiveSync*
PS >
```

Get a list of users and info regarding their device:

```
PS >
PS > Get-Mailbox -ResultSize:Unlimited | ForEach {Get-
ActiveSyncDeviceStatistics -Mailbox:$_.Identity} | ft
Identity,Devicemodel,DeviceOS,DeviceUserAgent,DeviceType,LastSucce
ssSync
PS >
```

Get a list of all devices and info regarding their users:

```
PS >
PS > Get-ActiveSyncDevice | select
UserDisplayName,DeviceID,DeviceType,DeviceUserAgent,DeviceModel,Na
me
PS >
```

To export any of the above commands into an Excel Document for reporting purposes, simply add the following to the end of any of the above commands to export it into a CSV file:

```
 | Export-CSV file.csv
```

http://www.matrix42.com