Security Removable Media Manager (secRMM)

Intune Access Control Setup Guide

Version 9.9.13.0
(May 2019)

Protect your valuable data
© 2011 Squadra Technologies, LLC. ALL RIGHTS RESERVED.

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Squadra Technologies, LLC.

If you have any questions regarding your potential use of this material, contact:

Squadra Technologies, LLC
7575 West Washington Ave
Suite 127-252
Las Vegas, NV 89128 USA
www.squadratechnologies.com
email: info@squadratechnologies.com

Refer to our Web site for regional and international office information.

TRADEMARKS

Squadra Technologies, secRMM are trademarks and registered trademarks of Squadra Technologies, LLC. Other trademarks and registered trademarks used in this guide are property of their respective owners.

Disclaimer

The information in this document is provided in connection with Squadra Technologies products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Squadra Technologies products. EXCEPT AS SET FORTH IN Squadra Technologies's TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, Squadra Technologies ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL Squadra Technologies BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF Squadra Technologies HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Squadra Technologies makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Squadra Technologies does not make any commitment to update the information contained in this document.

Squadra Technologies Administrator Guide
Created - March 2011
Contents

INTRODUCTION ........................................................................................................................................4

SETUP DECISIONS ...................................................................................................................................4

WHY USE INTUNE? .................................................................................................................................5
WHY USE SCCM? ....................................................................................................................................5

PREREQUISITES .......................................................................................................................................5

SETUP OVERVIEW ...................................................................................................................................5

AZURE INTUNE .......................................................................................................................................5
SCCM .......................................................................................................................................................5

SETUP DETAILS .......................................................................................................................................6

DOWNLOAD secRMMDEPLOYMENT.zip .................................................................................................6
AZURE ......................................................................................................................................................7
Setup 2 Azure “applications” ..................................................................................................................7
Login to your Azure tenant .......................................................................................................................7
Create Azure Native Application ...........................................................................................................7
Create Azure Web Api Application ........................................................................................................23
Giving the Azure Applications consent ..................................................................................................42
SCCM ......................................................................................................................................................44
CONFIGURE secRMM TO GET MOBILE DEVICE STATE ......................................................................51
Azure ......................................................................................................................................................52
Deploying the certificate .........................................................................................................................55
SCCM ......................................................................................................................................................55
LINK THE MOBILE DEVICE SERIAL NUMBERS TO AZURE ...............................................................55
Azure ......................................................................................................................................................56
SCCM ......................................................................................................................................................57

CONTACTING SQUADRA TECHNOLOGIES SUPPORT ........................................................................59

ABOUT SQUADRA TECHNOLOGIES, LLC. ..........................................................................................59
Introduction

Squadra Technologies security Removable Media Manager (secRMM) software is Windows security software that runs on your company’s workstations and servers. secRMM manages and monitors removable media. In this context, Removable media is defined as external hard disks, USB (flash) drives, smart phones, tablets, SD-Cards, CD-ROM and DVD. Such devices typically use the computers Universal Serial Bus (USB) ports to connect to the computer. Removable media devices are popular because they are very convenient when you want to copy files around or backup data. secRMM allows you to track all write activity to the removable media devices in your computer environment as well as giving you the ability to control (or authorize) who can write to the removable media devices.

This document is focused on removable media that is contained within mobile devices (either the devices flash storage or an SD-card). Mobile devices are so popular now that there are many software products which help organizations manage how mobile devices are used within the work place. These software products are called/categorized as “Mobile Device Management” (MDM) products. Microsoft has a MDM product named Intune that runs in the Microsoft cloud. Microsofts cloud is named Azure. All of the MDM products focus on security.

Unfortunately, when it comes to connecting the mobile device over a USB cable, the MDM products either allow or disallow a USB connection (i.e. either on or off). This is a sub-optimal solution for two reasons:

1. The device must be enrolled in the MDM to enforce this rule (i.e. either USB allowed or disallowed). This is roughly analogous to saying that police officers can catch all criminals but only if the criminals first go to the police station to get finger-printed. Otherwise it is not possible to catch them.
2. A rule that only enforces allowed or disallowed is on the one hand (allowed) too relaxed and on the other hand (disallowed) to restrictive. The right solution is to have policy such as secRMM to control read, write and who can have this access, from where they can copy data from, etc.

secRMM can be configured to use the mobile device definitions in Microsoft Intune or Microsoft System Center Configuration Manager (SCCM) to decide if a mobile device can be used over the USB connection. A mobile device can be used over a USB connection to transfer files to and from the mobile device to the Windows Desktop computer it is connected to (over the USB cable). You can configure secRMM to check if the mobile device is simply enrolled in Intune or that the device’s state (within the MDM) must be “compliant” before it can be used over the USB connection. Whether or not a mobile device is compliant is defined by the organization configuring the MDM and the devices within the MDM. An example would be that a device is compliant if the organizations apps were installed on the device.

If the functionality in the paragraph above is a desirable feature for your environment, this document will help you setup this secRMM feature.

Setup decisions

Since secRMM can provide this functionality using either Intune or SCCM, you must first decide which component (i.e. Intune or SCCM) that you will have secRMM use.
Why use Intune?

When secRMM uses Intune to get the mobile device state, it also gets more auditing data (if supplied within Intune) about the mobile device such as imei (International Mobile Equipment Identity), meid (Mobile equipment identifier), phone number, subscriber carrier (i.e. ATT, Verizon, etc.), Wi-Fi Mac address, etc. In addition, the Intune data is usually more recent than the data within SCCM. This is because the Intune data is transferred to SCCM within polling cycles whereas Intune updates its database based on when it communicates with the mobile device itself.

Why use SCCM?

When secRMM uses SCCM to get the mobile device state, the setup/configuration is simpler than when setting up for Intune. However, secRMM does not get as much auditing data about the mobile device as when it uses Intune. In addition, the mobile devices state could not reflect the current state of the mobile device since SCCM gets the mobile device data from Inutne using a polling method.

Prerequisites

Regardless of whether you use Intune or SCCM, you will always need to have a licensed Intune instance in Azure. By default, this also means you will have an Azure Active Directory (AAD) instance. Both Intune and AAD are defined within your Azure tenant. A tenant is a Microsoft term that can be thought of as a container that holds services, programs and data in the cloud that your company can access. Each tenant within Azure has a unique id (Microsoft calls this the “tenant id” and “directory id”).

Setup overview

Azure Intune

Here are the high-level steps we will take to setup the secRMM connection to your Azure tenant (i.e. AAD and Intune).

1. Setup 2 Azure “applications”
   a. Native
   b. Web API
2. Cache Web API credentials
3. Configure secRMM to get mobile device state from Intune
4. Link the mobile device serial numbers to mobile device definitions in Intune

SCCM

Here are the high-level steps we will take to setup the secRMM connection to your SCCM environment.

1. Setup SCCM/Intune connector within SCCM
2. Setup SCCM Cache Scheduled Task on SCCM site server
3. Configure secRMM to get mobile device state from SCCM
4. Link the mobile device serial numbers to mobile device definitions in SCCM

**Setup details**

**Download secRMMDeployment.zip**

Regardless of whether you configure Azure Intune or SCCM, there are files you will need that can be downloaded from the Squadra Technologies web site as shown in the screenshot below.

Please download and extract secRMMDeployment.zip file. These extracted files will be mentioned in the sections below.
Azure Setup 2 Azure “applications”

Of course, Microsoft does not allow external programs access to your Azure tenant by default. If you want to allow a program access to the services (i.e. AAD and Intune) within your Azure tenant, you must define an “application” within your Azure tenant (via your AAD). The external program must go through these Azure “applications” to access the services and data within Azure. These Azure “applications” are really nothing more than a collection of security settings that tell Azure what parts of Azure the external program can access. There are two different kinds of “applications”: 1. Native and 2. “Web API”. As of the writing of this document, Azure (IMHO) has some cleanup on how this all works but this document will walk you through the process so you really do not need to have a deep understanding of the whys and hows. Once you follow the steps, the end result will be that secRMM can access the mobile device data in your AAD and Intune so that secRMM can make the decision about mounting your mobile devices over USB connections based on your Azure tenant data.

Login to your Azure tenant

Using a web browser, go to URL https://portal.azure.com. You will need to supply your Azure userid and password. The userid you use must be defined as the Azure Global Administrator account. Azure Global Administrators are the only userids that can define Azure “applications”.

Create Azure Native Application

Once you are logged in, you will be at your Azure tenant Dashboard. On the left hand side of the web page, find and select “Azure Active Directory”. A new column will appear just to the right of the Dashboard column.
In the new column, select “App registrations”. A new page will appear just to the right of the “App registrations” column. In the new page, at the top, select “New application registration”.

A form will appear that wants you to specify the “Application name”, the “Application type” and a “Redirect URI” as shown in the screenshot below. For the “Application name”, type `secRMMIntuneNativeApp` (although this is a free form text field and can have any value you want, we recommend you specify `secRMMIntuneNativeApp` so the documentation below will match your environment). For the “Application type”, select “Native”. For the “Redirect URI”, type `http://www.squadratechnologies.com/Products/secRMM/Azure/Intune/secRMMApplicationConsent.aspx`. Now click the Create button.
Azure AAD will create the application when you click the Create button and will show you the new application as shown in the screenshot below. We will need to use the “Application ID” on this screen when we configure secRMM later in this document. Next, click the “Settings” button (in the upper left hand corner) as shown in the screenshot below.

When you click the “Settings” button, you will see a new column to the right of the Application information named “Settings”.

We now need to click the “Owners” in the “Settings” column.
When you click the “Owners” button, you will see a new column to the right of the Settings column named “Owners”. In the “Owners” column, you will click the “Add owner” button as shown in the screenshot below.

When you click the “Add owner” button, a new column named “Add owner” will appear to the right of the Owners column as shown in the screenshot below. In the list of userids, select your AAD Global Administrator account (the one you should be logged into) and then click the “Select” button.
When you click the “Select” button, you will end up back on the “Owners” column as shown in the screenshot below. On the “Owners” column, click the “Close” (i.e. the X in the upper right hand corner) button.

When you click the “Close” button, you will end up back on the “Settings” column as shown in the screenshot below. On the “Settings” column, click the “Required permissions” row.
When you click the “Required permissions”, a new column will appear. There will only be one row named “Windows Azure Active Directory”. Click this row (i.e. the row named “Windows Azure Active Directory”).

When you click the row named “Windows Azure Active Directory”, another column will appear named “Enable Access”. In this column, we will uncheck the “Sign in and read user profile” and then check “Access the directory as the signed-in user” and “Read and write directory data” as shown in the screenshot below. The last step on the “Enable Access” column is to click the “Save” button (at the top-left of the column) as shown in the screenshot below.
When you click the “Save” button, you will end up back on the “Required permissions” column as shown in the screenshot below. On the “Required permissions” column, click the “Add” button.

When you click the “Add” button, another column will appear named “Add API access”. On the “Add API access” column, click the “Select an API” row as shown in the screenshot below.
When you click the “Select an API” row, another column will appear named “Select an API”. On the “Select an API” column, click the “Microsoft Graph” row and then the “Select” button as shown in the screenshot below.
When you click the “Select” button, another column will appear named “Enable Access”. On the “Enable Access” column, you will click the “access permissions” (shown in the screenshot below) and then the "Select" button as shown in the screenshot below.
When you click the "Select" button, you will end up back on the "Add API access" column as shown in the screenshot below. On the "Add API access" column, click the "Done" button.

When you click the "Done" button, you will end up back on the "Required permissions" column as shown in the screenshot below. On the "Required permissions" column, click the "Add" button again.
When you click the “Add” button, another column will appear named “Add API access”. On the “Add API access” column, click the “Select an API” row as shown in the screenshot below.

When you click the “Select an API” row, another column will appear named “Select an API”. On the “Select an API” column, click the “Microsoft Intune API” row and then the “Select” button as shown in the screenshot below.
When you click the “Select” button, another column will appear named “Enable Access”. On the “Enable Access” column, you will click the “access permissions” and then the “Select” button as shown in the screenshot below.
When you click the “Select” button, you will end up back on the “Add API access” column as shown in the screenshot below. On the “Add API access” column, click the “Done” button.

When you click the “Done” button, you will end up back on the “Required permissions” column as shown in the screenshot below. On the “Required permissions” column, click the “Grant Permissions” button.

When you click the “Grant Permissions” button, a pop-up window will appear asking if you want to grant the permissions for the secRMMIntuneNativeApp as shown in the screenshot below. Click the “Yes” button.
When you click the “Yes” button, you will end up back on the “Required permissions” column as shown in the screenshot below. On the “Required permissions” column, click the “Close” (i.e. the X in the upper right hand corner) button.

<table>
<thead>
<tr>
<th>API</th>
<th>APPLICATION PERM</th>
<th>DELEGATED PERM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Azure AD</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Microsoft Graph</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Microsoft Intune API</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

When you click the “Close” button, you will end up back on the “Settings” column as shown in the screenshot below. On the “Settings” column, click the “Close” (i.e. the X in the upper right hand corner) button.
When you click the “Close” button, you will end up back on the main column for the secRMM\IntuneNativeApp as shown in the screenshot below. On this column, click the “Manifest” button.

When you click the “Manifest” button, you will see the “Edit Manifest” column appear to the right as shown below. In the manifest, you need to change the 4th line value from true to false.

Now click the “Save” button on the top left of the “Edit manifest” column as shown in the screenshot below.
When you click the “Save” button, the manifest will be saved as shown in the screenshot below. On the “Edit manifest” column, click the “Close” (i.e. the X in the upper right hand corner) button.

When you click the “Close” button, you will end up back on the main column for the secRMMIntuneNativeApp as shown in the screenshot below. On this column, click the “Close” (i.e. the X in the upper right hand corner) button.
When you click the “Close” button, you will end up back on the main “App registrations” column and the app you just created from the steps above will be listed.

Create Azure Web Api Application

In the exact same way we created the native application in the previous section, in the “App registrations” column, at the top, select “New application registration”.

![Image of a web page showing the creation of an Azure Web Api Application registration]
A form will appear that wants you to specify the “Application name”, the “Application type” and a “Redirect URI” as shown in the screenshot below. For the “Application name”, type secRMMIntuneWebApp (although this is a free form text field and can have any value you want). For the “Application type”, select “Web app / API”. For the “Redirect URI”, type http://www.squadratechnologies.com/Products/secRMM/Azure/Intune/secRMMApplicationConsent.aspx. Now click the Create button.
Azure AAD will create the application when you click the Create button and will show you the new application as shown in the screenshot below. We will need to use the “Application ID” on this screen when we configure secRMM later in this document. Next, click the “Settings” button (in the upper left hand corner) as shown in the screenshot below.

When you click the “Settings” button, you will see a new column to the right of the Application information named “Settings”.

We now need to click the “Owners” in the “Settings” column.
When you click the “Owners” button, you will see a new column to the right of the Settings column named “Owners”. In the “Owners” column, you will click the “Add owner” button as shown in the screenshot below.

When you click the “Add owner” button, a new column named “Add owner” will appear to the right of the Owners column as shown in the screenshot below. In the list of userids, select your AAD Global Administrator account (the one you should be logged into) and then click the “Select” button.
When you click the “Select” button, you will end up back on the “Owners” column as shown in the screenshot below. On the “Owners” column, click the “Close” (i.e. the X in the upper right hand corner) button.

When you click the “Close” button, you will end up back on the “Settings” column as shown in the screenshot below. On the “Settings” column, click the “Required permissions” row.
When you click the “Required permissions”, a new column will appear. There will only be one row named “Windows Azure Active Directory”. Click this row (i.e. the row named “Windows Azure Active Directory”).

When you click the row named “Windows Azure Active Directory”, another column will appear named “Enable Access”. In this column, we will uncheck the “Sign in and read user profile” (in the Delegated Permissions section, towards the bottom of the column) and then check “Read directory data” and “Read and write devices” (in the Application Permissions section, at the top of the column) as shown in the screenshot below. The last step on the “Enable Access” column is to click the “Save” button (at the top-left of the column) as shown in the screenshot below.
When you click the “Save” button, you will end up back on the “Required permissions” column as shown in the screenshot below. On the “Required permissions” column, click the “Add” button.

1. uncheck

When you click the “Save” button, you will end up back on the “Required permissions” column as shown in the screenshot below. On the “Required permissions” column, click the “Add” button.
When you click the “Add” button, another column will appear named “Add API access”. On the “Add API access” column, click the “Select an API” row as shown in the screenshot below.

When you click the “Select an API” row, another column will appear named “Select an API”. On the “Select an API” column, click the “Microsoft Graph” row and then the “Select” button as shown in the screenshot below.
When you click the “Select” button, another column will appear named “Enable Access”. On the “Enable Access” column, you will click the “access permissions” (shown in the screenshot below) and then the "Select" button as shown in the screenshot below.
When you click the “Select” button, you will end up back on the “Add API access” column as shown in the screenshot below. On the “Add API access” column, click the “Done” button.
When you click the "Done" button, you will end up back on the "Required permissions" column as shown in the screenshot below. On the "Required permissions" column, click the "Add" button again.

When you click the “Add” button, another column will appear named “Add API access”. On the “Add API access” column, click the “Select an API” row as shown in the screenshot below.
When you click the "Select an API" row, another column will appear named "Select an API". On the "Select an API" column, click the "Microsoft Intune API" row and then the "Select" button as shown in the screenshot below.
When you click the “Select” button, another column will appear named “Enable Access”. On the “Enable Access” column, you will click the “access permissions” (as shown in the screenshot below) and then the “Select” button as shown in the screenshot below.
When you click the “Select” button, you will end up back on the “Add API access” column as shown in the screenshot below. On the “Add API access” column, click the “Done” button.
When you click the “Done” button, you will end up back on the “Required permissions” column as shown in the screenshot below. On the “Required permissions” column, click the “Grant Permissions” button.

When you click the “Grant Permissions” button, a pop-up window will appear asking if you want to grant the permissions for the secRMMIntuneWebApp as shown in the screenshot below. Click the “Yes” button.

When you click the “Yes” button, you will end up back on the “Required permissions” column as shown in the screenshot below. On the “Required permissions” column, click the “Close” (i.e. the X in the upper right hand corner) button.
When you click the “Close” button, you will end up back on the “Settings” column as shown in the screenshot below. On the “Settings” column, click the “Close” (i.e. the X in the upper right hand corner) button.

When you click the “Close” button, you will end up back on the main column for the secRMMIntuneWebApp as shown in the screenshot below. On this column, click the “Manifest” button.
At this point, we need to create a certificate for secRMM\IntuneWebApp and then add it to the Manifest. To do this, you will keep your browser open (editing the Manifest) and perform the steps below on your local computer:

1. In the step above where you downloaded the secRMM\Deployment.zip, when you extract the zip, there is a directory named MDM\Intune. In that directory is a powershell script named CreateManifest.ps1. The script is listed below for your convenience. The CreateManifest.ps1 file should be in a temporary directory on your local hard drive. For example, we will use C:\temp.

   ```powershell
   $cert=New-SelfSignedCertificate -Subject "CN=secRMM\IntuneWebApp" -CertStoreLocation "Cert:\LocalMachine\My" -KeyExportPolicy Exportable -KeySpec Signature
   $bin = $cert.RawData
   $base64Value = [System.Convert]::ToBase64String($bin)
   $bin = $cert.GetCertHash()
   $base64Thumbprint = [System.Convert]::ToBase64String($bin)
   $keyid = [System.Guid]::NewGuid().ToString()
   $jsonObj = @
   @{
       customKeyIdentifier=$base64Thumbprint;
       keyId=$keyid;
       type="AsymmetricX509Cert";
       use="Verify";
       value=$base64Value
   }
   $keyCredentials=ConvertTo-Json @($jsonObj) | Out-File "keyCredentials.txt"
   ```

2. Open a CMD window
3. Change directory (CD) into the temporary directory where you stored the file named CreateManifest.ps1 from step 1 above. Example: `cd C:\temp`
4. Type `Powershell` and hit enter
5. Type `.\CreateManifest.ps1` and hit enter
6. Type `exit`
7. Type `notepad keyCredentials.txt`
8. In notepad, use `Edit->Select All`
9. In notepad, use `Edit->Copy`

We will replace the "[ ]" on the keyCredentials line with the contents of the clipboard text we set in step 9 above as shown in the two screenshots below. Note that the values will be different from the values in the second screenshot below. When you have pasted in the keyCredentials text, click the Save button in the upper left hand corner.
{
  "appId": "b83d22eb-67f3-4284-a2b6-65251fc77860",
  "appRoles": [],
  "availableToOtherTenants": false,
  "displayName": "secRMMIntuneWebApp",
  "errorUrl": null,
  "groupMembershipClaims": null,
  "optionalClaims": null,
  "acceptMappedClaims": null,
  "homepage": "http://www.squadratechnologies.com/Pro",
  "informationalUrls": {
    "privacy": null,
    "termsOfService": null
  },
  "identifierUris": [
    "https://anthonysquadratechnologies.onmicrosoft.c",
  ],
  "keyCredentials": [],
  "knownClientApplications": [],
  "logoutUrl": null.
On the "Edit Manifest" column, click the "Close" (i.e. the X in the upper right hand corner) button.

When you click the “Close” button, you will end up back on the main column for the secRMMIntuneWebApp as shown in the screenshot below. On this column, click the “Close” (i.e. the X in the upper right hand corner) button.
When you click the “Close” button, you will end up back on the main “App registrations” column and the app you just created from the steps above will be listed.

The setup tasks within Azure are now completed. Please go to the section below titled “Configure secRMM to get mobile device state”.

**Giving the Azure Applications consent**

The last step we must perform is to give “Azure consent” to the 2 Azure applications you created above. This means we must create 2 urls and paste them into the browser, one at a time. The url is mostly the same except for the application ids which are unique from each other. The other value you will need is your Azure domain name which usually ends with “.onmicrosoft.com”. These 2 changes are shown in the url below.

https://login.microsoftonline.com/your_Azure_domain_name/adminconsent?client_id=the_Azure_application_id&state=123456&redirect_uri=http%3a%2f%2fwww.squadratechnologies.com%2fProducts%2fsecRMM%2fAzure%2fIntune%2fsecRMMApplicationConsent.aspx

When you paste the url into the browser and hit enter, you will login as your Azure Global domain administrator account. Once logged in, you will see a dialog similar to the screenshot below. Click the accept button.
When the consent operation completes, you will see a web page as shown in the screenshot below. The `admin_consent` value should be equal to True.
secRMM "Azure application consent" result:

admin_consent=True
tenant=805a3915-a0ac-4d2a-9ec8-4bb702169c30
state=123456

SCCM

The first step in configuring SCCM is to connect SCCM to Intune using the Microsoft SCCM/Intune connector. The steps to perform this are well documented by Microsoft at https://technet.microsoft.com/en-us/library/jj884158.aspx.
The next step is to create a scheduled task on the SCCM site server. The scheduled task will create a cache file. This cache file is used by secRMM when an end-user "USB connects" a mobile device to their Windows workstation (or server). The files used by the scheduled task are located in the secRMMDeployment.zip file that you downloaded and extracted at the beginning of this document.

Within the extracted secRMMDeployment.zip, there will be a directory named \MDM\SCCM. Within that directory are 3 files:

1. `secRMMSCCMIntuneConnectorDataPull.xml` – the scheduled task that can be imported by the Windows Task Scheduler User Interface
2. `secRMMSCCMMDMScheduledTask.cmd` – this is the cmd file called by the Windows Task Scheduler (i.e. the Action performed by the Scheduled Task)
3. `secRMMSCCMMDMScheduledTask.sql` – this is the SQL query that the scheduled task uses to read from the SCCM database

On the SCCM site server, create a directory structure named `C:\SCCM_MDM_Cache\secRMMSCCMMDMScheduledTask`. Put the 3 files listed above into this directory as shown in the screenshot below.
Edit file secRMMSCCMMDMScheduledTask.cmd and change lines 5-7 as shown in the screenshot below.

Line 5 should be the computer name that is running the SQL instance where your SCCM database is located. Line 6 is the name of your SCCM database. This is always in the form of CM_XXX where XXX is the 3 character SCCM identifier. Line 7 is where the cache file will be written to.

```bash
# Begin batch script
@echo off

SET SQLServerAndInstance=localhost
SET DatabaseName=CM_SQT
SET MDMEnrollmentCache=C:\SCCM_MDM_Cache

REM Purpose: Get the mobile device data out of the SCCM database and
REM copy it to the cache location (specified in the secRMM
REM RequireMDMEnrollment property).
REM
REM Copyright (c) 2018 Squadra Technologies
REM
REM ******************************************************************************
```

Go into the Windows “Task Scheduler” and select “Import Task”.

---

Page 46
Select `secRMMSCCMIntuneConnectorDataPull.xml` (which we extracted in the steps above) as shown in the screenshot below.

![Task Scheduler with Import Task highlighted](image1)

Change the userid to a user that has read access to the SCCM database as shown in the screenshot below.

![Open dialog box with secRMMSCCMIntuneConnectorDataPull.xml highlighted](image2)

Change the userid to a user that has read access to the SCCM database as shown in the screenshot below.
On the "Triggers" tab, you can modify how often the scheduled task will execute.
When you are done performing any modification to the scheduled task, click the OK button as shown in the screenshot below.
Provide the password for the userid as shown in the screenshot below.
You can verify that the schedule task is functioning properly by running it from the Task Scheduler. The output will be a file named C:\SCCM_MDM_Cache\secRMMCacheSCCM.txt. This file holds the mobile devices extracted from the SCCM database (by way of the SCCM/Intune connector).

The setup tasks within SCCM are now completed. Please go to the section below titled “Configure secRMM to get mobile device state”.

**Configure secRMM to get mobile device state**

Now that we have setup where secRMM will get the mobile device state (i.e. either Intune or SCCM), we will go back to our on-premise domain (where secRMM is deployed) to give secRMM the information about how to get the mobile device state data. secRMM can be centrally managed with either System Center Configuration Manager (SCCM) or Active Directory Group Policy Objects (AD GPO). You can also individually manage a single computer using the Windows “Computer Management” MMC interface (a good tool when you are testing...before you deploy a policy to your entire environment).

Regardless of which interface you use to configure secRMM, there is a secRMM property named “RequireMDMEnrollment”. Double click the “RequireMDMEnrollment” row to open the window that lets you configure the Intune or SCCM connection. As you can see in the screenshot, there are several options available to you. We will break down each option below.
The first checkbox (labeled “On”) is required to be checked. It is here to be consistent with all of the other secRMM on/off properties. The second checkbox (labeled “Enforce when device is plugged in.”) will make secRMM communicate with Intune or SCCM as soon as the end-user connects the mobile device using the USB cable. If you do not check this checkbox, secRMM will enforce the rule when an end-user tries to transfer a file to the mobile device. Next, is the drop-down listbox (labeled “Required state”). There are two options: Enrolled and Compliant. This tells secRMM what state is required to allow the mobile device to be used over a USB connection. The compliant state is the most strict since the mobile device must be both enrolled and compliant at the same time. It is not possible for the device to be compliant if it is not enrolled.

Regardless of whether you are using Azure Intune or SCCM, you must always specify a Cache directory. This is the last field on the “Require MDM Enrollment” dialog. Notice that this must be a network share. Make sure that you give Read/Write permissions to all “Domain computers”.

We have now come to the point where we can tell secRMM to get the mobile device state from either Azure Intune or SCCM. Follow the appropriate subsection below based on whether you are using Azure Intune or SCCM.

**Azure**

We need to use the “Require MDM Enrollment” dialog 2 times. The first time is to specify the information required to access the Azure Web App. Then, we will change the values to access the Azure Native App. We will leave the values to access the Azure Native App as the default values. For the Azure Web App values, we will put them into the cached directory in a txt file. This is described below.

For both configurations, we will need to have the Azure TenantID. You get this value from your Azure portal as shown in the screenshot below. Within Azure, the TenantID is listed as the “Directory ID”.
There is a script file we will use that is located in the secRMMDeployment.zip file that you downloaded and extracted at the beginning of this document. Within the extracted secRMMDeployment.zip, there will be a directory named \MDM\Intune. Within that directory there is a file named SaveIntuneCredentialsToFile.cmd. We are going to configure the Azure Web App in the secRMM “Computer Management” MMC, save it and then run SaveIntuneCredentialsToFile.cmd so that we can save the “RequireMDMEnrollment” values (for the Azure Web App) to a file that will be stored in the cache directory.

To connect to the Azure Web App, we need to supply 4 values:

1. **Method** = **Certificate**
2. **TenantID** = this is the “Directory ID” you got from the Azure portal (see above)
3. **ApplicationID** = this is listed in the Azure portal->Azure Active Directory->App registrations->secRMMIntuneWebApp
4. **Certificate = CN=secRMMIntuneWebApp**

Now, click the “Test” button to make sure the connection succeeds. A success message will look like the screenshot below. If the test is unsuccessful, you can look in “C:\Program Files\secRMM\AdminUtils\MDM\Intune\secRMMMDMIntune.log for the errors.

Once you have a successful connection, save the secRMM values by clicking the OK button on the “Require MDM Enrollment” dialog. Once the values are saved, run the SaveIntuneCredentialsToFile.cmd. This will create a file named secRMMMDMCertificateCredentials.txt. Copy secRMMMDMCertificateCredentials.txt to the Cache directory that you specified in the “Require MDM Enrollment” dialog.

Now, we will double click the RequireMDMEnrollment row in the secRMM MMC again to change the values to be for the Azure Native App (i.e. use a UserId/Password). To connect to Azure Intune (for the Azure Native App), we need to supply 5 values:

1. Method – select UserIdPassword
2. TenantID – specify your Azure TenantID
3. ApplicationID – specify the Azure **Native** Application ID that was generated from the steps above (note, be sure you change from the Web Application ID to the Native Application ID)
4. UserID – specify the Azure Global Administrator ID
5. Password – specify the Azure Global Administrator ID password

Once you specify all the required values, click the “Test” button to ensure that the connection to Azure Intune is valid. Once you have a successful connection, click the OK button on the “Require MDM Enrollment” dialog to save the values.

**Deploying the certificate**

Since the Azure Intune method requires a certificate, you must distribute the certificate to the computers running secRMM within your environment. Please use the Microsoft documentation at https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/deployment/distribute-certificates-to-client-computers-by-using-group-policy to perform the distribution of the certificate.

**SCCM**

The only thing you need to do for SCCM is to select the “Use SCCM Intune connector” checkbox and to specify the Cache.

**Link the mobile device serial numbers to Azure**

It is now time to link the mobile device serial number from secRMM (i.e. the true serial number on the mobile device firmware) to the definition of the mobile device within Azure Intune or SCCM. To do this, secRMM comes with a user interface program located in “C:\Program Files\secRMM\AdminUtils\MDM\Intune” named secRMMLinkMobileDevices.exe. You must right-mouse click this program and select “Run as Administrator”.
**Azure**

In the secRMMLinkMobileDevices program, click the “Connect to Intune” button in the upper left hand corner. A dialog with title “Connect to Intune” will appear. The values in this dialog will be pre-populated with the values from the secRMM “RequireMDMEnrollment” that you supplied earlier. Therefore, just click the OK button.
The program will go to Azure Intune and retrieve the mobile device definitions and populate column 1 named “Device Name”. Column 2 (named secRMM Serial Number) will be empty (unless you have already supplied a value previously). The get the "secRMM Serial Number", attach the mobile device via a USB cable. Then, go into the secRMM event log (or use the secRMM Excel AddIn, secRMM Reports or any other means to get the secRMM event 400 for that specific mobile device) and copy the serial number specified for that mobile device and paste it into column 2. When you paste (or manually type) the serial number, be sure to tab out of the cell. When you tab out of the cell, that invokes the storing of the value into Azure Intune.

**SCCM**

In the secRMMLinkMobileDevices program, click the “Connect to SCCM” button in the upper left hand corner. A dialog with title “Connection Properties” will appear. Specify the server name where the SCCM database resides and specify the SCCM database name (CM_xxx where xxx is the 3 character site code). Make sure that the userid you are logged in on has read and write permissions to the SCCM database.
The program will go to the SCCM database and retrieve the mobile device definitions and populate column 1 named “Device Name”. Column 2 (named secRMM Serial Number) will be empty (unless you have already supplied a value previously). The get the “secRMM Serial Number”, attach the mobile device via a USB cable. Then, go into the secRMM event log (or use the secRMM Excel AddIn, secRMM Reports or any other means to get the secRMM event 400 for that specific mobile device) and copy the serial number specified for that mobile device and paste it into column 2. When you paste (or manually type) the serial number, be sure to tab out of the cell. When you tab out of the cell, that invokes the storing of the value into Azure Intune.
Contacting Squadra Technologies Support

Squadra Technologies Support is available to customers who have purchased a commercial version of secRMM and have a valid maintenance contract or who are in a trial mode of the product.

When you contact Support please include the following information:

1. The version of secRMM you have installed.
2. The Windows versions you have installed: XP, 2003 Server, 2008 Server R2, Vista, Windows 7, etc.
3. Whether the Windows Operating System is 32bit or 64bit.
4. The specific issue you are contacting support for.

About Squadra Technologies, LLC.

Squadra Technologies delivers innovative products that help organizations get more data protection within the computer infrastructure. Through a deep expertise in IT operations and a continued focus on what works best, Squadra Technologies is helping customers worldwide.

Contacting Squadra Technologies, LLC.

Phone 562.221.3079 (United States and Canada)
Email info@squadratechnologies.com
Mail Squadra Technologies, LLC.
World Headquarters
7575 West Washington Ave. Suite 127-252
Las Vegas, NV 89128
USA
Web site http://www.squadratechnologies.com/