

# SID Mismatch Re-Enrollment Process

## Goal

To automatically un-enroll and re-enroll a Windows 10 device in the case where the enrollment SID does not match the logged in user SID. Since we don't yet support multi-user, this results in only partial management of the device. Several things stop working such as App Sampling and Cert Sampling. Because App Sampling fails, this causes the SFD agent to not automatically upgrade itself on console upgrades. It will also prevent any user context profiles or apps from being deployed.

## Change-Log

v2.3 - Jun 29, 2020

- Added disabling/re-enabling toast notifications for silent un-enrollment and re-enrollment process
- Renamed file to be WS1-ReEnroll.ps1

v2.3 - March 5, 2020

- Added in PSADT class for querying logged in active user
- Changed enrollment check logic to check for positive enrollment vs non-valid enrollment
- added ping to Workspace ONE server before running
- Some updates to logging text and bug fixes
- Changed parameter from UPN to Username

v2.1 - Mar 3, 2020

- Fixed issue with renaming old log files
- Added additional logging info when enrolling via HUB

v2 - Feb 28, 2020

- Added 5 min wait after oma-dm removal to ensure everything is removed properly
- Added logic to re-name hub logs after removal of hub

## Files

1. Enrollment Batch file WS1-ReEnroll.bat, and WS1-ReEnroll.ps1  
<https://github.com/vmware-samples/euc-samples/tree/master/Windows-Samples/Product%20Provisioning/Re-Enroll%20Workspace%20ONE%20-%20SID%20Mismatch>
2. Airwatch Agent (get correct version matching customer console or download latest from getwsone.com)

## Pre-reqs:

- Has 4 required parameters. These are used for the silent enrollment command line:
  - Server
  - LGName (Org Group ID)
  - Username
  - Password
- A logged in user does have to be detected. If no logged in user is detected the script will exit.

## Usage

- Note this does require 64bit powershell
  - 64bit - powershell -executionpolicy bypass -file .\WS1-ReEnroll.ps1 -server [ws1uem.awmdm.com](https://www.airwatch.com) -lgname staging -username [staging@staging.com](mailto:staging@staging.com) -password 11111
  - 32bit - %WINDIR%\Sysnative\WindowsPowerShell\v1.0\powershell.exe -executionpolicy bypass -file .\WS1-ReEnroll.ps1 -server [ws1uem.awmdm.com](https://www.airwatch.com) -lgname staging -username [staging@staging.com](mailto:staging@staging.com) -password 11111
- Logfile is saved: C:\ProgramData\Airwatch\UnifiedAgent\Logs\WS1-ReEnroll.log

## Script Process:

1. Sets variables
2. Checks for elevation. If not run with elevation privileges it will exit.
3. Tests log path (function), if not found will create it
4. Gets current architecture. If 32bit: exits. If 64: continues.
5. Checks agent path (function Check-Agent-Path). Script expects AirwatchAgent.msi to be in same directory as script. If not found, will download from getwsone.com
6. Enrollment Check (function Enrollment-Check) - verifies if there is a valid MDM enrollment and returns UPN
  - a. Get GUID here: HKLM:\SOFTWARE\Microsoft\Provisioning\OMADM\Accounts\\*
  - b. Checks other registry keys to ensure that there is a valid enrollment
    - i. if Valid, returns the UPN email of enrolled user
    - ii. if null or enrollment state is not valid, returns \$false
7. Checks windows SID (function Check-SID)

- a. Uses a C# class from PSADT to detect current logged in user.
- b. If valid, then compares logged in user SID with the enrollment SID. Does additional logic to ensure that the MDM SID its comparing is really one of ours (ProviderID = AirwatchMDM).
- c. If windows SID and Enrollment SID match, return \$true, else return \$false
8. If Enrollment-Check and Check-SID both return \$true, then script knows there already is a healthy enrollment with correctly matching SIDs and exits.
9. if one or both are false,
  - a. Disables toast notifications so the end user doesn't see a scary message about Work resources being removed from their PC
  - b. Calls Uninstall-Hub function
    - i. Uninstall-Hub searches for any installed Hub GUID and ADA(software distribution client) Guid and uninstalls them. This should also trigger MDM unenrollment, but in case it doesn't we call DeviceEnroller.exe to force a sync which then will trigger un-enrollment since Hub is missing
  - c. Enroll-Hub function is called and does command line enrollment with values from the parameters passed to it as well as ASSIGNTOLOGGEDINUSER=Y parameter. It then waits 5 min for enrollment to complete and then does another Enrollment-Check.

## Create a Product in WS1

Prep:

1. Download the batch file above and update the content with your environment specific details. Example:

```
cd %~dp0
%WINDIR%\Sysnative\WindowsPowerShell\v1.0\powershell.exe -executionpolicy bypass -file .\WS1-ReEnroll.ps1 -Server ds1380.awmdm.com -
LGName bpeppin -Username username -Password N0tReal
```

2. Download or get the correct version of Intelligent hub (Airwatch Agent.msi)
3. Download WS1-ReEnroll.ps1 file

## Create Files/Actions

1. Go to Devices > Provisioning > Components > Files/Action
2. Click "Add Files/Actions"
3. General tab - fill out basic details

### Edit Files/Actions

General	Files	Manifest
Name *	WS1-Enroll-SID-Check	
Description		
Version	3	
Platform	Windows Desktop	
Managed By *	VMware bpeppin	

4. Files Tab. Upload the 3 files and specify target path (such a C:\temp)

General	Files	Manifest																
<div>ADD FILES</div> <table border="1"> <thead> <tr> <th>File Name</th> <th>Path</th> <th>Version</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>Enroll_Cn1380.bat</td> <td>C:\Temp\Enroll_Cn1380.bat</td> <td>1.0</td> <td>Local</td> </tr> <tr> <td>Enroll-WS1.ps1</td> <td>C:\Temp\Enroll-WS1.ps1</td> <td>1.0</td> <td>Local</td> </tr> <tr> <td>AirwatchAgent.msi</td> <td>C:\Temp\AirwatchAgent.msi</td> <td>1.0</td> <td>Local</td> </tr> </tbody> </table>			File Name	Path	Version	Type	Enroll_Cn1380.bat	C:\Temp\Enroll_Cn1380.bat	1.0	Local	Enroll-WS1.ps1	C:\Temp\Enroll-WS1.ps1	1.0	Local	AirwatchAgent.msi	C:\Temp\AirwatchAgent.msi	1.0	Local
File Name	Path	Version	Type															
Enroll_Cn1380.bat	C:\Temp\Enroll_Cn1380.bat	1.0	Local															
Enroll-WS1.ps1	C:\Temp\Enroll-WS1.ps1	1.0	Local															
AirwatchAgent.msi	C:\Temp\AirwatchAgent.msi	1.0	Local															

5. Manifest.
  - a. Click Add Action

- b. Run, System, Path to your batch file

## Edit Manifest

Action(s) To Perform *	<input type="text" value="Run"/>
Execution Context *	<input type="text" value="System"/>
Command Line and Arguments to run *	<input type="text" value="C:\Temp\Enroll_Cn1380.bat"/>
TimeOut (-1 for infinite) *	<input type="text" value="0"/> ⓘ

6. Click Save

## Create Product

1. Go to Devices > Provisioning > Product List View and click "Add Product"
2. General Tab - fill out and assign smart group

## Edit Product

<div>General Manifest Conditions Deployment Dependencies</div>	
Name *	<input type="text" value="WS1-Enroll-SID-Check"/> ⓘ
Description	<input type="text"/>
Managed By *	<input type="text" value="VMware bpeppin"/>
Smart Groups	<div><div>✱ All Devices (VMware bpeppin) ✕</div><div>Start typing to add a group 🔍</div></div>
	<div>VIEW DEVICE ASSIGNMENT</div>
Assignment Rules	<div>ADD RULES</div>

3. Manifest Tab. Click "Add" and select "Install Files/Action". Select the Files/Action item you just created.

## Add Manifest

Action(s) To Perform \*

Install Files/Actions

Files/Actions \*

WS1-Enroll-SID-Check




### Edit Product



General **Manifest** Conditions Deployment Dependencies

+ ADD



Up	Down	Step Number	Action Type	Persistent	Description	
▲	▼	1	Install Files/Actions	No	Files/Actions = WS1-Enroll-SID-Check	  

Items 1-1 of 1

4. Conditions tab - leave default

5. Deployment Tab - change Product Type to "Elective". This will require you to manually push the product to devices ad-hoc.

### Edit Product

General Manifest Conditions **Deployment** Dependencies

Server Date and Time : 2/26/2020 9:21 AM

Activation Date

M/D/YYYY

12:00 AM

Deactivation Date

M/D/YYYY

12:00 AM

Pause/Resume



Product Type

Elective

6. Dependencies Tab - leave default

7. Click save and Activate the product.


## Deploy to Device


1. Go to a device and on device details page click on More > Products.

2. Select the product and click Send

bpeppin2 CLIENT1  
Virtual Machine | 10.0.17134 | Ownership: Corporate - Dedicated

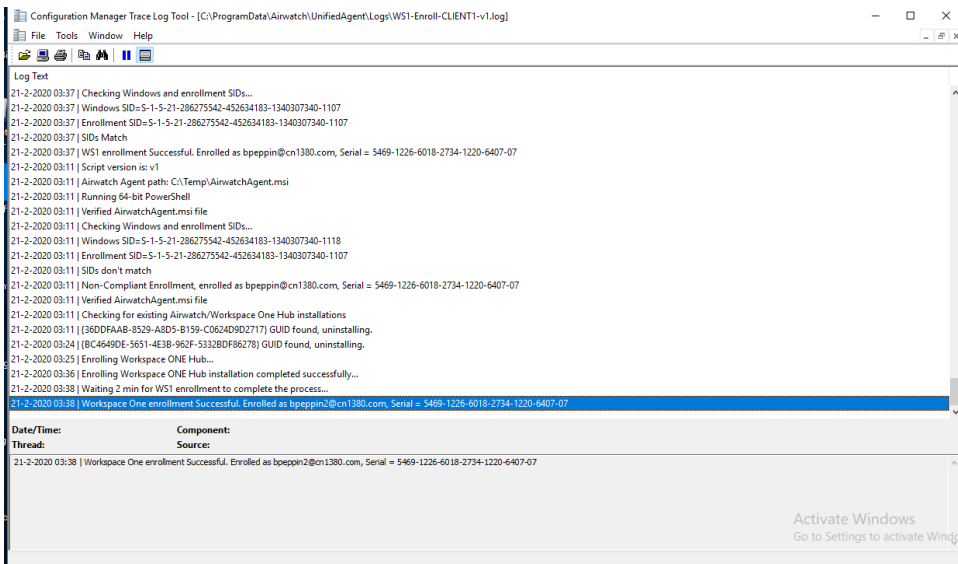
Summary Compliance Profiles Baselines Apps Updates Content Location User **Products**

Q JOBS  EXPORT Search List

Q VIEW HISTORY 

Name	Product Set	Status	Type	Last Job ID	Date	Last Job Status
Hub 2001 Targeted Upgrade		Compliant	Elective	54407	2/24/2020 2:11:04 PM	Completed
Save-Logs		Non-Compliant - MustPush	Elective	N/A	N/A	N/A
WS1 Health Check		Non-Compliant - MustPush	Elective	N/A	N/A	N/A
WS1-Enroll-SID-Check		Non-Compliant - MustPush	Elective	N/A	N/A	N/A

- Use the refresh button to check status. Since you included the AiwatchAgent.msi this might take a little longer to run since downloading this file directly from DS servers is slow.
- On client, check the log: "C:\ProgramData\Airwatch\UnifiedAgent\Logs\WS1-Enroll-[hostname]-v1.log". You can see the test machine I have here where I run it both from a matching SID and mis-matching SID domain accounts.



## Checking for mis-matching SID using Sensors

### Create Sensors

We can create 3 sensors to check the environment for mismatching SID to get an idea of how many are affected

Create each sensor under Devices > Provisioning > Custom Attributes > Sensors. The examples below give sensor details and config details.

- get\_windows\_sid - [https://github.com/vmware-samples/euc-samples/blob/master/Windows-Samples/Sensors/get\\_windows\\_sid.ps1](https://github.com/vmware-samples/euc-samples/blob/master/Windows-Samples/Sensors/get_windows_sid.ps1)
- get\_enrollment\_sid - [https://github.com/vmware-samples/euc-samples/blob/master/Windows-Samples/Sensors/get\\_enrollment\\_sid\\_32\\_64.ps1](https://github.com/vmware-samples/euc-samples/blob/master/Windows-Samples/Sensors/get_enrollment_sid_32_64.ps1)
- check\_sid\_mismatch - [https://github.com/vmware-samples/euc-samples/blob/master/Windows-Samples/Sensors/Check\\_Matching\\_SID\\_Sensor\\_32\\_64.ps1](https://github.com/vmware-samples/euc-samples/blob/master/Windows-Samples/Sensors/Check_Matching_SID_Sensor_32_64.ps1)

Assign to your device and run "query sensors" to force them to run. Note, if a user is logged off, sensors can't be manually triggered. They will run on agent check in schedule (usually every 4 hours).

### Create Intelligence Report

- Launch WS1 Intelligence
- Go to Reporting > Reports. Add Report
- Category: Workspace ONE UEM > Device Sensors.
- Rename the report. Example "Check SID Mismatch"
- Under Filters select "sid\_mismatch" and either select available data or use "start with: s". Add the other columns as well:

Filters

sid\_mismatch starts with s

sid\_mismatch Starts With s

CLOSE

Report Preview

This report preview has 2 records. Refreshed a few seconds ago

EDIT COLUMNS

get_enrollment_sid_workspace	get_windows_sid	sid_mismatch	device_guid
S-1-S-21-3652684359-1046837282-3359684849-1002	S-1-S-21-3652684359-1046837282-3359684849-1002	SID_Match	4f0a955f-9961-4a5f-9836-465c059608d
S-1-S-21-286275542-452634183-1340307340-1107	S-1-S-21-286275542-452634183-1340307340-1107	SID_Mismatch	732ba655-9708-4bc1-b633-b59b6979500