

Product Platform: **Windows**
Product Version: **All versions**
Document Date: **8 August 2025**
Document Version: **1.0**
Classification: **Public**

Using PowerShell to Query ABR

Introduction

The Admin By Request API allows you to get the necessary data into your preferred SIEM system. This blog covers how to test functionality and get data from Admin By Request using Windows PowerShell.

In order to use *Invoke-RestMethods* cmdlets used during this task, you need to be running Windows PowerShell version **3.0** or higher.

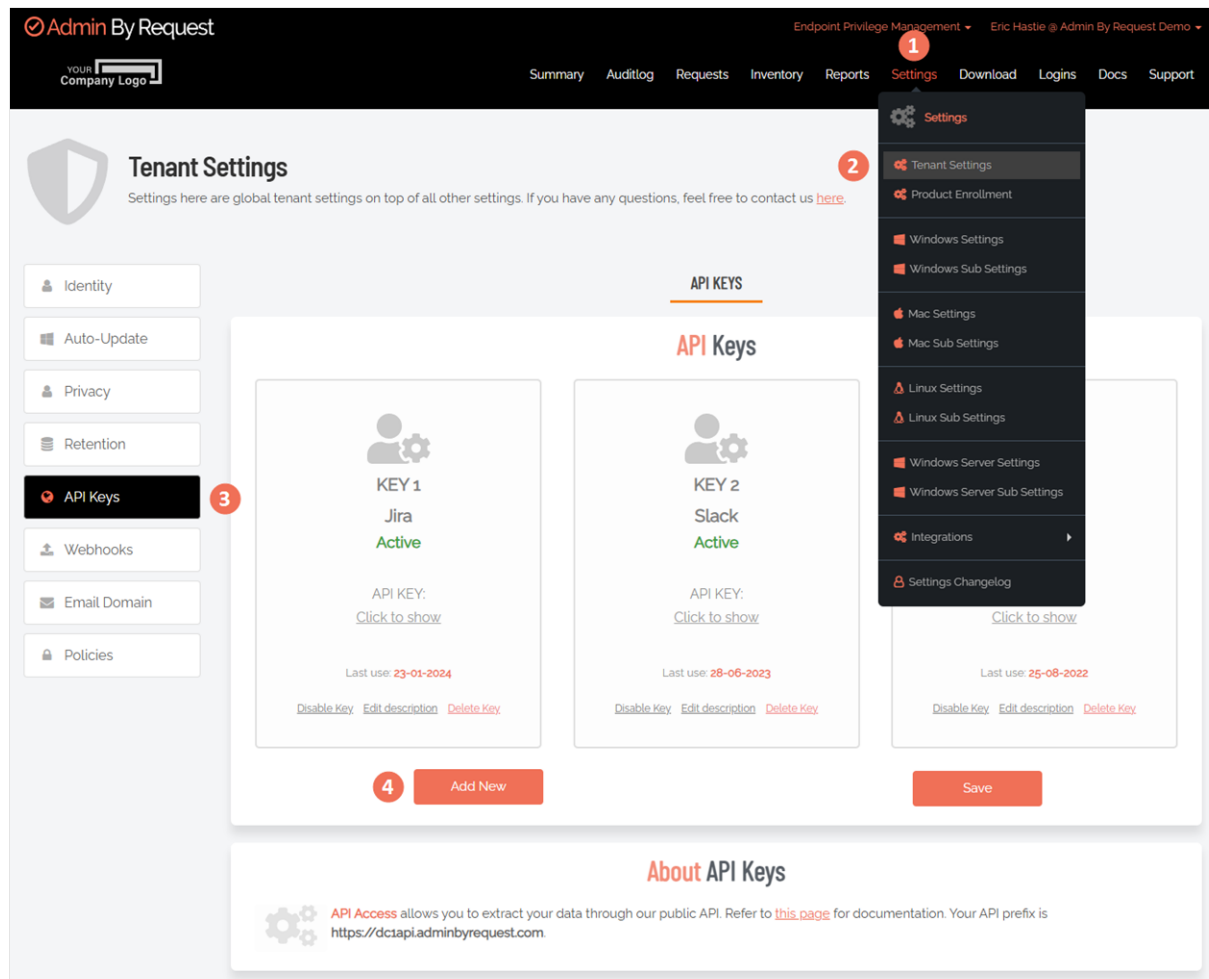
Procedure

There are five tasks involved:

- "Enable and Copy API Key" on the next page
- "Copy Required URLs from Resources" on the next page
- "Start PowerShell and Declare API Key" on page 4
- "Define General Variables" on page 4
- "Get Data" on page 5

A. Enable and Copy API Key

1. In the Admin By Request portal, navigate to menu **Settings > Tenant Settings > API Keys > API KEYS**:



2. If there is no key available, click button **Add New**. If there is a key available, select **Click to show** for the relevant key, followed by **Copy to clipboard**.
3. Paste the API Key into notepad (or similar) to be retrieved later, so that it is not overwritten in the next step.

B. Copy Required URLs from Resources

This task locates and copies two URLs used to make queries in subsequent tasks.

NOTE

- For this example, we want to return the current inventory and the latest auditlog data.
- The full URL depends on your data center. This example uses *dc1api* (**https://dc1api.adminbyrequest.com** in the URL). You might see a different data center in your full URL (e.g. *dc2api*).

Primary method

This method requires JavaScript to be enabled on your endpoint.

1. Copy Inventory URL

- In the portal, go to page [Inventory API](#).
- From the list of resources, click to copy the *inventory* URL:

Inventory API		
This page explains how to get your inventory data extracted. Note that the example array of computers further down shows only the first computer and a subset of software installed for readability. You can use query parameters to filter your search.		
Resources		
URL	Description	Method
https://dc1api.adminbyrequest.com/inventory	Returns an array of inventory computers	GET
https://dc1api.adminbyrequest.com/inventory/{id}	Returns one computer's inventory by id	GET
https://dc1api.adminbyrequest.com/inventory/{computername}	Returns one computer's inventory by computer name	GET
https://dc1api.adminbyrequest.com/inventory/{id}	Delete one computer by id	DELETE
https://dc1api.adminbyrequest.com/inventory/{computername}	Delete one computer by computer name	DELETE

You might need to select and copy rather than simply clicking the URL.

- Paste the inventory URL into notepad (or similar) to be retrieved later, so that it is not overwritten in the next step.

2. Copy Auditlog URL

- In the portal, go to page [Auditlog API](#).
- From the list of resources, click to copy the *auditlog* URL:

Auditlog API		
This page explains how to get your auditlog data extracted. Note that the example array of audit log entries further down shows only the first entry and a subset of scan results for readability. You can use query parameters to filter your search.		
Resources		
URL	Description	Method
https://dc1api.adminbyrequest.com/auditlog	Returns an array of auditlog entries	GET
https://dc1api.adminbyrequest.com/auditlog/{id}	Returns one auditlog entry	GET
https://dc1api.adminbyrequest.com/computers/{computername}/auditlog	Returns an array of auditlog entries for a certain computer	GET
https://dc1api.adminbyrequest.com/users/{user}/auditlog	Returns an array of auditlog entries for a certain user (user account of full name)	GET
https://dc1api.adminbyrequest.com/auditlog/delta	Returns an array of changed auditlog entries since last call (see further down)	GET

You might need to select and copy rather than simply clicking the URL.

- Paste the auditlog URL into notepad (or similar) to be retrieved later, so that it is not overwritten in the next step.

Alternative method

This works whether JavaScript is enabled or not. Confirm your data center as described below, then append **/inventory** and **/auditlog** respectively at steps 1 and 2 above.

To determine your data center, go to page [Tenant Settings > API Keys](#) in the portal and check which API prefix is shown under **About API Keys**. The data center (which is also the API prefix) will be one of the following:

- <https://dc1api.adminbyrequest.com>**(Europe - Netherlands)
- <https://dc2api.adminbyrequest.com>**(USA)
- <https://dc3api.adminbyrequest.com>**(UK)

- <https://dc4api.adminbyrequest.com>(Europe - Germany)
- <https://dc6api.adminbyrequest.com>(Asia)

Make a note of your prefix - among other things, this is the domain used when an API Key is created.

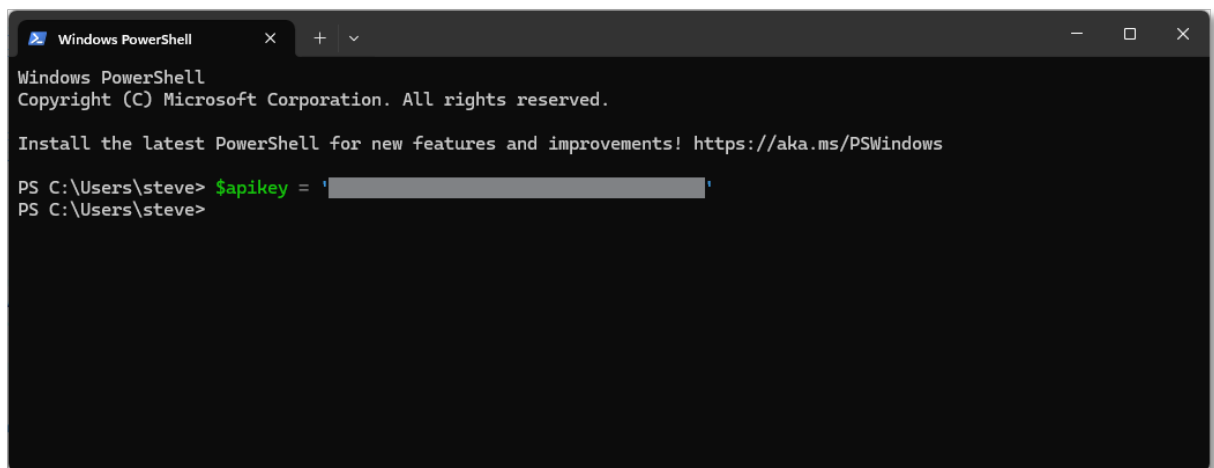
C. Start PowerShell and Declare API Key

If you want to run the code within this blog as a script, you might need to change the default execution policy in PowerShell to *bypass* or *unrestricted* using the following line of code:

```
set-executionpolicy bypass -scope process
```

An explanation of why this is required is outside the scope of this blog - refer to [PowerShell's Execution Policy](#) for more information.

1. Launch Windows PowerShell and declare the API Key by copying and pasting the following line of code into the window:
`$apikey = '<your-api-key>'`
2. Replace **<your-api-key>** in this line of code with the API Key you copied in ["Enable and Copy API Key" on page 2](#).
3. Press **Enter**:



D. Define General Variables

This task defines several variables to make the code easier to work with.

1. Define a *header variable* by copying and pasting the following line of code into the window:
`$header = @{"apikey"=$apikey}`
2. Press **Enter**:
3. Define an *inventory variable* by copying and pasting the following line of code into the window:
`$inventory = 'https://dclapi.adminbyrequest.com/inventory'`
(Replace the URL in this line of code with the inventory URL you copied in ["Copy Inventory URL" on the previous page](#))
4. Press **Enter**.

5. Define an *auditlog* variable by copying and pasting the following line of code into the window:

```
$auditlog = 'https://dclapi.adminbyrequest.com/auditlog'
```

(Replace the URL in this line of code with the auditlog URL you copied in "Copy Auditlog URL" on page 3)

6. Press **Enter**.

The outcome of this task (and the \$apikey variable) is as follows:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\steve> $apikey = '...'
PS C:\Users\steve> $header = @{"apikey"=$apikey}
PS C:\Users\steve> $inventory = 'https://dclapi.adminbyrequest.com/inventory'
PS C:\Users\steve> $auditlog = 'https://dclapi.adminbyrequest.com/auditlog'
PS C:\Users\steve> |
```

E. Get Data

1. Get Inventory Data

- a. Copy and paste the following line of code into the PowerShell window:

```
Invoke-Restmethod -uri $inventory -header $header -Method GET
```

```
PS C:\Users\steve> Invoke-Restmethod -uri $inventory -header $header -Method GET

id           : 56537538
name         : Roses-Mac
inventoryAvailable : True
inventoryDate : 2025-07-16T07:51:26
abrClientVersion : 5.2.0
abrClientInstallDate : 2025-04-09T16:06:45
notes       :
user        : @{account=localadmin; fullName=Local Admin; email=; phone=; isAdmin=True; domain=; isDomainJoined=; isAzureJoined=; orgUnit=; orgUnitPath=; groups=}
owner       : @{account=rosesmith; fullName=Rose Smith}
computer    : @{domain=NZOFFICE; isDomainJoined=; isAzureJoined=; orgUnit=Computers; orgUnitPath=Computers; groups=; localAdmins=System.Object[]; users=System.Object[]}
operatingSystem : @{platform=Apple; platformCode=1; name=macOS 14.7; version=Version 14.7.1 (Build 23H222); release=; build=; buildUpdate=; type=; typeCode=0; bits=64; installDate=2024-06-06T00:00:00}
hardware     : @{make=Apple Inc.; model=VMware20.1; type=Desktop; typeCode=1; serviceTag=VMware-56 4d 2a 08 3e 30 d8 a5-c4 06 c7 28 a5 af 9; diskSize=85; diskFree=60; diskStatus=Not Supported; memory=16384; noMonitors=1; monitorResolution=1920x1200; bitlockerEnabled=False; isCompliant=; tpmEnabled=; tpmVersion=}
network      : @{publicIP=122.253.208.32; privateIP=192.168.200.134; macAddress=08:0c:29:3d:3e:a2; nicSpeed=4294 mbit; hostName=}
location     : @{city=Lower Hutt; region=Unknown; country=New Zealand; latitude=-41.1975; longitude=174.9189; googleMapsLink=https://maps.google.com/?q=-41.1975,174.9189; hourOffset=}
software     :

id           : 56885268
name         : WIN10-VM2
inventoryAvailable : True
inventoryDate : 2025-08-12T11:42:26.45
abrClientVersion : 8.6.1
abrClientInstallDate : 2025-08-12T11:42:13
notes       :
user        : @{account=steved; fullName=; email=; phone=; isAdmin=False; domain=NZOFFICE; isDomainJoined=False; isAzureJoined=False; orgUnit=; orgUnitPath=; groups=}
owner       : @{account=STEVED; fullName=}
computer    : @{domain=NZOFFICE; isDomainJoined=True; isAzureJoined=False; orgUnit=; orgUnitPath=; groups=; localAdmins=System.Object[]; users=System.Object[]}
operatingSystem : @{platform=Windows; platformCode=0; name=Windows 10 Enterprise Evaluation; version=; release=1803; build=17134; buildUpdate=2208; type=Workstation; typeCode=0; bits=64; installDate=2025-07-17T00:00:00}
hardware     : @{make=VMware, Inc.; model=VMware20.1; type=Desktop; typeCode=1; serviceTag=VMware-56 4d 2a 08 3e 30 d8 a5-c4 06 c7 28 a5 af 9; cpu=13th Gen Intel Core i7-13780H; cpuSpeed=2918; cpuCores=4; diskSize=63; diskFree=42; diskStatus=OK; memory=8588; noMonitors=1; monitorResolution=1024x768; bitlockerEnabled=False; isCompliant=; tpmEnabled=False; tpmVersion=}
network      : @{publicIP=101.98.189.2; privateIP=192.168.200.131; macAddress=08:0c:29:af:90:0e; nicSpeed=1000 mbit; hostName=}
location     : @{city=Auckland; region=Unknown; country=New Zealand; latitude=-36.821400; longitude=174.706900; googleMapsLink=https://maps.google.com/?q=-36.821400,174.706900; hourOffset=12}
software     :
```

2. Get Auditlog Data

- Copy and paste the following line of code into the PowerShell window:

`Invoke-Restmethod -uri $auditlog -header $header -Method GET`

```
PS C:\Users\steve> Invoke-Restmethod -uri $auditlog -header $header -Method GET

install      : {{application=XProtect; version=; vendor=}}
uninstall    : {}
elevatedApplications : {}
scanResults  : {}
id           : 300798384
traceNo      : 281511150
settingsName : Global
type         : Run As Admin
typeCode     : 0
status       : Finished
statusCode   : 2
application  : @{{file=XProtect.app; path=/Library/Apple/System/Library/InstallerSandboxes/.PKInstallSandboxManager-SystemSoftware/55CD1037-734A-4EDF-86CA-88942500BF39.activeSandbox/Root/Library/Apple/System/Library/CoreServices; name=xprotect; vendor=; version=; sha256=; scanResult=Clean; scanResultCode=0; threat=; virustotalLink=https://www.virustotal.com/latest-scan/; preapproved=False}}
user         : @{{account=ROOT; fullName=root; email=; phone=; isAdmin=True}}
computer     : @{{name=Roses-Mac; platform=Mac; platformCode=1; make=Apple Inc.; model=VMware 20.1}}
reason       :
approvedBy   :
approvedByEmail :
deniedReason :
deniedBy     :
deniedByEmail :
ssoValidated : False
requestTime  : 2025-07-16T07:48:42
requestTimeUTC : 2025-07-15T19:48:42
startTime    : 2025-07-16T07:48:42
startTimeUTC  : 2025-07-15T19:48:42
endTime      : 2025-07-16T07:48:42
endTimeUTC    : 2025-07-15T19:48:42
responseTime  :
auditlogLink  : https://www.adminbyrequest.com/AuditLog?Page=AppElevations&ID=281511150&ShowFilter=false

install      : {}
uninstall    : {}
elevatedApplications : @{{name=Windows Command Processor; path=C:\Windows\System32; file=cmd.exe; version=10.0.26100.1 (WinBuild.160101.0800); vendor=Microsoft Corporation; sha256=875B19DD825BF2086119A0A43431002FB2A3DAFEE7F755453C4DA9109D070566; scanResult=Clean; scanResultCode=0; threat=; virustotalLink=https://www.virustotal.com/latest-scan/875B19DD825BF2086119A0A43431002FB2A3DAFEE7F755453C4DA9109D070566}, @{{name=IP Configuration Utility; path=C:\Windows\System32; file=ipconfig.exe; version=10.0.26100.1 (WinBuild.160101.0800); vendor=Microsoft Corporation; sha256=7953A5C5A4B778BFD13C13CEE8CDB7CD876A1F8106B31D94D640724511D884EE; scanResult=Clean; scanResultCode=0; threat=; virustotalLink=https://www.virustotal.com/latest-scan/7953A5C5A4B778BFD13C13CEE8CDB7CD876A1F8106B31D94D640724511D884EE}, @{{name=TCP/IP Ping Command; path=C:\Windows\System32; file=PING.EXE; version=10.0.26100.1 (WinBuild.160101.0800); vendor=Microsoft Corporation; sha256=96F2ABAC2542F4CD59628D14AF1F1935FEBFA56C75C3430E108B5465EBC823E; scanResult=Clean; scanResultCode=0; threat=; virustotalLink=https://www.virustotal.com/latest-scan/96F2ABAC2542F4CD59628D14AF1F1935FEBFA56C75C3430E108B5465EBC823E}}
```

To output the data to a TXT or CSV file for further aggregation, simply add **>> filename** to the end of the command.

For example:

```
Invoke-Restmethod -uri $auditlog -header $header -Method GET >>
auditlog.txt
```

Voila! You have now successfully used Windows PowerShell to get inventory and auditlog data written to screen or file.

Acknowledgment

This procedure was created with the assistance of [Mads Christian Mozart Johansen](#).