POWERGUARD Cloud – Powershell transcription solution for Entra Joined devices

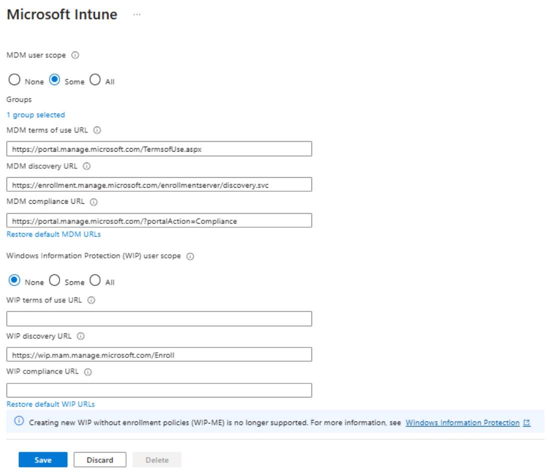
Contents

Prerequisites

- Endpoints should be managed by MDM. In our document we will use Intune, but you can use also other MDM. If you need to register your endpoint to intune use the (Optional) How to enroll endpoints to Intune section.
- Make sure your endpoint is entra-joined: run **DSREGCMD /STATUS** and look for: **AZUREADJOINED: YES**
- Users should have Intune license. The current recommended license for POC is Intune plan 1.

(optional) How to enroll endpoints to Intune

- **Make sure your enrolling user has an Intune-capable license** (examples: Microsoft 365 E3/E5, EMS E3/E5, Intune standalone). I will use Microsoft Intune Plan 1 in this document.
- **Set Intune as the MDM authority (if needed)**
    - Go to Intune admin center.
    - Navigate to Tenant administration → Intune enrollment.
    - If prompted to set the MDM authority, select Microsoft Intune
    - For MDM user scope select “Some” if you want to test individual user or “All” of you want to deploy it on the entire organization (simplest for lab).
        - **Note** - “Some” is a cleaner enterprise pattern because it prevents “accidental enrollment” of all users/devices.
    - In case you choose “Some”, choose “No group selected” and add your security group contain the relevant user for the test process.
    - **Note** – in enrollment to Intune require reboot.
- Validate enrollment - Confirm device shows in Intune.
    - Go to Intune admin center.
    - Devices → All devices
    - Find the device name.
    - Confirm Managed by: Intune / co-managed.
    - Once done, choose “Save”.



Microsoft Intune

MDM user scope: None [✓ Some] All

Groups: 1 group selected

MDM terms of use URL: https://portal.manage.microsoft.com/TermsofUse.aspx

MDM discovery URL: https://enrollment.manage.microsoft.com/enrollmentserver/discovery.svc

MDM compliance URL: https://portal.manage.microsoft.com/?portalAction=Compliance

Restore default MDM URLs

Windows Information Protection (WIP) user scope: [✓ None] Some All

WIP terms of use URL:

WIP discovery URL: https://wip.mam.manage.microsoft.com/Enroll

WIP compliance URL:

Restore default WIP URLs

Creating new WIP without enrollment policies (WIP-ME) is no longer supported. For more information, see Windows Information Protection

Save Discard Delete

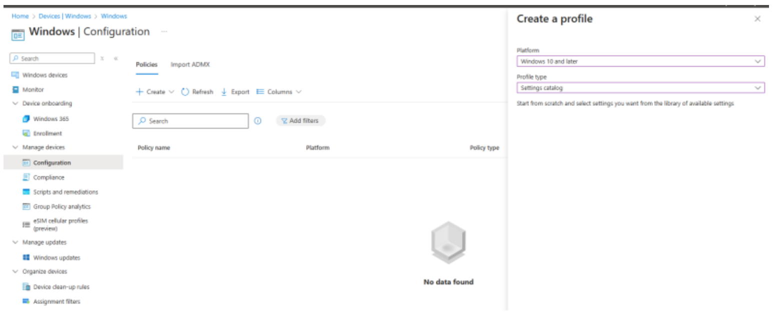
Enroll the Entra-joined device

Here you have two reliable paths:

1. **Most direct – Windows settings > enroll only in device management.**
    - <u>We will use this method in our lab environment.</u>
2. **User-friendly - Company Portal**. Require Intune company portal installation from Microsoft Store.

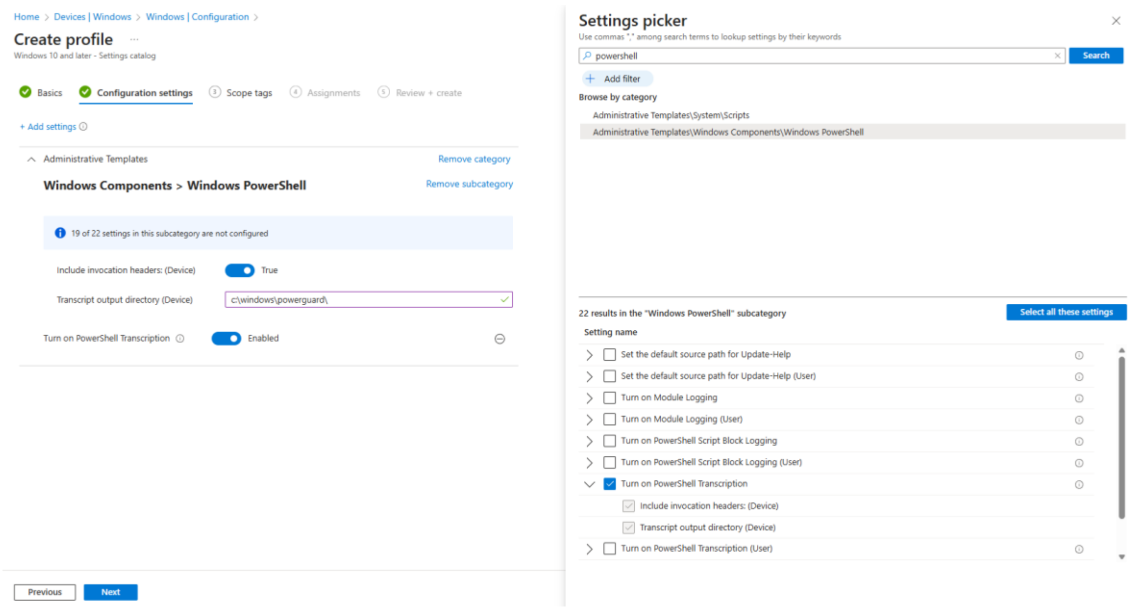
Enable PowerShell Transcription (Intune-managed)

- ➢ Create Intune Configuration Profile
- ➢ Intune admin center > Devices > Windows > manage devices > Configuration > Create > new policy
    - o Platform: Windows 10 and later
    - o Profile type: Settings catalog.


Home > Devices | Windows > Windows
Windows | Configuration
Policies | Import ADMX
Policy name | Platform | Policy type
No data found

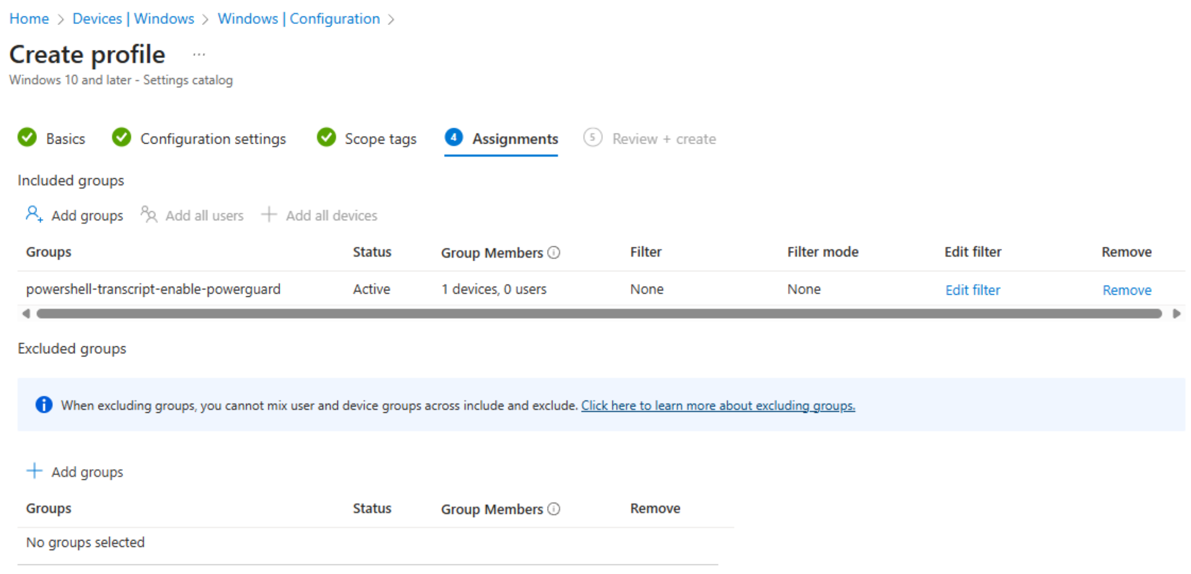
Create a profile
Platform: Windows 10 and later
Profile type: Settings catalog
Start from scratch and select settings you want from the library of available settings

- ➢ On create profile menu, give it a name, like “Enable PowerShell Transcription”, then choose next.
- ➢ On Configuration settings, on the right menu search for PowerShell, then enable “Turn on PowerShell Transcription”. Then, on the left menu, enable “include invocation headers: (Device)”, and “Turn on PowerShell Transcription”. On transcription output directory (Device) mention the output path for the PowerShell transcription log files.
- ➢ Once done, choose next.


Home > Devices | Windows > Windows | Configuration >
Create profile
Windows 10 and later - Settings catalog
Basics | Configuration settings | 3 Scope tags | 4 Assignments | 5 Review + create
Administrative Templates
Windows Components > Windows PowerShell
19 of 22 settings in this subcategory are not configured
Include invocation headers: (Device): True
Transcript output directory (Device): c:\windows\powerguard\
Turn on PowerShell Transcription: Enabled

Settings picker
Use commas "," among search terms to lookup settings by their keywords
powershell
Browse by category
Administrative Templates\System\Scripts
Administrative Templates\Windows Components\Windows PowerShell
22 results in the "Windows PowerShell" subcategory
Setting name
☐ Set the default source path for Update-Help
☐ Set the default source path for Update-Help (User)
☐ Turn on Module Logging
☐ Turn on Module Logging (User)
☐ Turn on PowerShell Script Block Logging
☐ Turn on PowerShell Script Block Logging (User)
☑ Turn on PowerShell Transcription
  ☑ Include invocation headers: (Device)
  ☑ Transcript output directory (Device)
☐ Turn on PowerShell Transcription (User)

- ➢ On Scope tags choose next unless its relevant for your testing scope.
- ➢ On assignment menu, under included groups, choose to “add groups” and add the security group that contains the relevant endpoint you want to enable Powershell transcription on. Then choose next.


Home > Devices | Windows > Windows | Configuration >

Create profile ...
Windows 10 and later - Settings catalog

Basics · Configuration settings · Scope tags · 4 Assignments · 5 Review + create

Included groups

Add groups · Add all users · Add all devices

Groups	Status	Group Members	Filter	Filter mode	Edit filter	Remove
powershell-transcript-enable-powerguard	Active	1 devices, 0 users	None	None	Edit filter	Remove

Excluded groups

When excluding groups, you cannot mix user and device groups across include and exclude. Click here to learn more about excluding groups.

Add groups

Groups	Status	Group Members	Remove
No groups selected			

- Accept the review and choose **Create**.

Enforce Folder Security

Why?

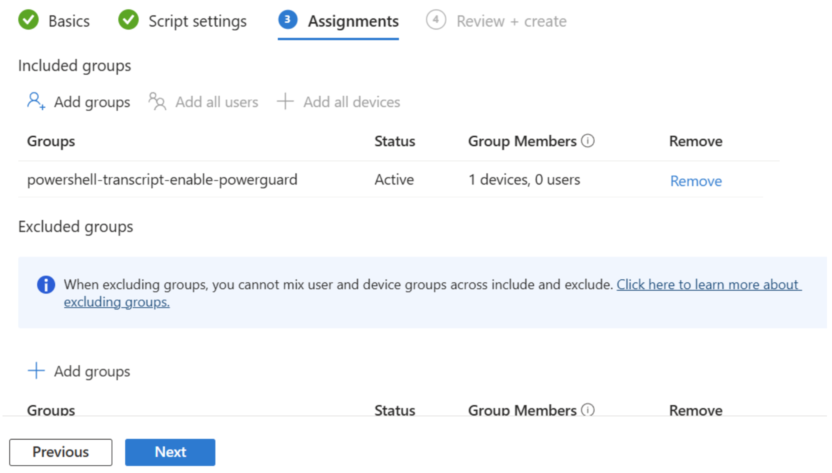
PowerShell transcripts may contain – credentials, tokens, command payloads, post-exploitation artifacts, etc.
If the transcript folder is writable/readable by standard users, an attacker can delete evidence, tamper with logs, read sensitive material.

Policy without filesystem hardening = incomplete control.

To create an Intune PowerShell Script:

- In Microsoft Intune admin center, navigate to Devices > Scripts and remediations > Platform scripts > Add > windows 10 and later.
- Name the script, for example: “PowerGuard - Secure PS Transcription Folder”, then choose next.
- Upload the powershell script provided by tenroot team (powerguardcloud_aclenforcement(MDM).ps1)
- On script settings, set the following:
  - Run this script using the logged on credentials: No
  - Enforce script signature check: No
  - Run script in 64-bit PowerShell: Yes
- On Assignment page, choose to assign it to groups, and choose the same security group contains the relevant endpoints that inherit the enabled powershell transcript configuration (in our case – “powershell-transcript-

enable-powerguard”). Then choose next.



Included groups

Groups	Status	Group Members	Remove
powershell-transcript-enable-powerguard	Active	1 devices, 0 users	Remove

Excluded groups

When excluding groups, you cannot mix user and device groups across include and exclude. Click here to learn more about excluding groups.

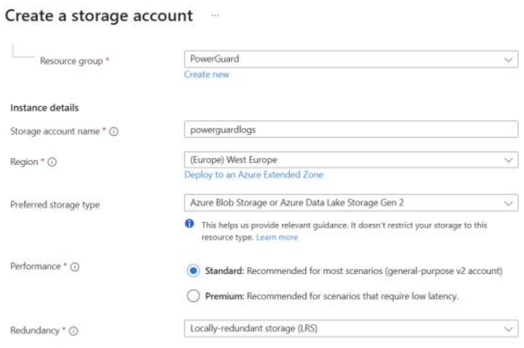
Groups	Status	Group Members	Remove

- ➢ On review + create page, choose to create the powershell script deployment.
- ➢ The script should execute in 5–15 minutes, Worst case: up to ~60 minutes (if device is idle, network not connected, etc. to enforce the sync immediately, on the windows machine go to settings > accounts > access work or school. Select your entra account, click info, then click sync.
- ➢ To validate the permissions inherent from intune, run icacls C:\windows\PowerGuard\
  you should see:
  - o SYSTEM: Full
  - o Administrators: Full
  - o Users: Read (or none, if you removed it)

Export PowerShell Transcripts to Azure (HTTPS, Blob-based)

Step 1 – create azure storage.

- ➢ In azure portal, open storage center (storage accounts)
- ➢ Choose to create storage account.
- ➢ Choose your subscription and resource group where the storage will be created.
- ➢ Give the storage account a name, for example: “powerguardlogs”
- ➢ Choose your relevant region, then choose for standard performance, LRS (locally-redundant storage) redundancy. Then choose next.

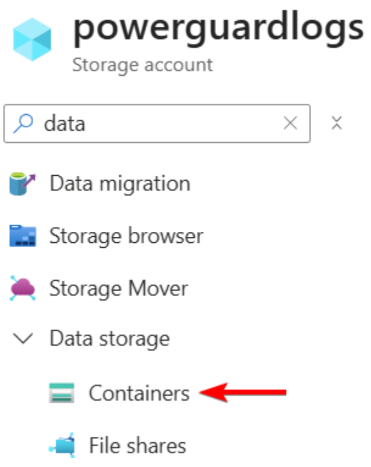

Create a storage account

Resource group *	PowerGuard
	Create new

Instance details

Storage account name *	powerguardlogs
Region *	(Europe) West Europe
	Deploy to an Azure Extended Zone
Preferred storage type	Azure Blob Storage or Azure Data Lake Storage Gen 2
	This helps us provide relevant guidance. It doesn't restrict your storage to this resource type. Learn more
Performance *	● **Standard:** Recommended for most scenarios (general-purpose v2 account)
	○ **Premium:** Recommended for scenarios that require low latency.
Redundancy *	Locally-redundant storage (LRS)

- On advanced page make sure “Require secure transfer for REST API operations” is checked to verify file transfer will use https and reject http request. Then choose next.
- On network page, make sure “Enabled from all networks” is selected. If restricted to selected networks, the scanner will fail with "Service Unavailable".
- Choose next and once you get to the Review + create page, choose Create.
- Once the storage account is ready, you need to create container. Enter the dedicated page of the storage account you just created (powerguardlogs), then choose under Data storage menu, choose Containers:



- Choose Add container to create a new container.
- For name, enter a name like: “powerguard-ps-transcript”, then choose Create.

generating the SAS token on this container

Important note - With Intune Platform scripts, you cannot achieve “zero plaintext exposure” of a SAS (Shared access signature) token on the endpoint. Intune does not provide a native secret vault for PowerShell scripts. Any SAS token used directly by the endpoint must exist in memory at runtime, and therefore can be extracted by a local admin.

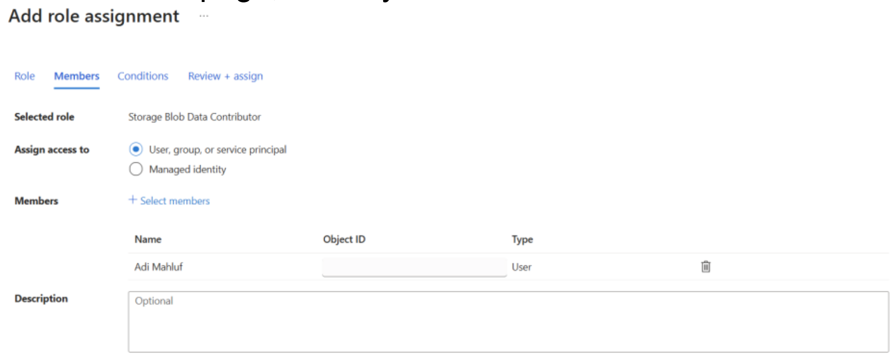
Security objectives

- Endpoint users (standard users) cannot view the token.
- Token is write-only.
- Token is short-lived.
- Token is scoped in one container.
- Token can be revoked without touching endpoints.
- No secrets hardcoded for long-term use

Option A — Short-lived, write-only SAS (recommended for lab / POC)

Pre-requisites:

- To grant write and add content access you must have the following permission on the storage account or container:
  - Storage Blob Data Owner
- To set the permission on relevant scope only:
  - Navigate to storage account > your storage account (powerguardlogs) > Access Control (IAM).
  - Choose +Add > Add role assignment.
  - Search for “**Storage Blob Data Owner**” and choose next.
  - On Members page, select your account then choose next



Selected role: Storage Blob Data Contributor

Assign access to: (●) User, group, or service principal ( ) Managed identity

Members:

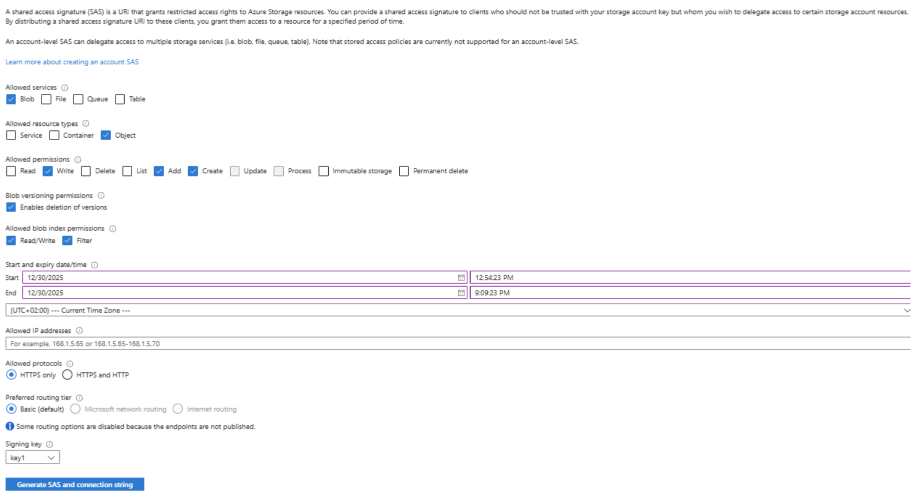
Name	Object ID	Type
Adi Mahluf		User

Description: Optional

  - On Review + assign choose Review + assign

Genrate the token

1. In Azure portal, navigate to storage account (powerguardlogs) > containers > powerguard-ps-transcripts > settings > **Shared access tokens**.
2. for allowed services – choose **Blob** only.
3. for allowed resource type – choose **Object** only.
4. for allowed permissions – choose only **Create**, **Write**, **Add**. Remove any other permissions.
5. For start and expiry date/time, set a time slot of 8 hours for your testing / POC.
6. On allowed protocols make sure “HTTPS only” is selected.
7. Once done, choose Generate SAS and connection string.



A shared access signature (SAS) is a URI that grants restricted access rights to Azure Storage resources. You can provide a shared access signature to clients who should not be trusted with your storage account key but whom you wish to delegate access to certain storage account resources. By distributing a shared access signature URI to these clients, you grant them access to a resource for a specified period of time.

An account-level SAS can delegate access to multiple storage services (i.e. blob, file, queue, table). Note that stored access policies are currently not supported for an account-level SAS.

Learn more about creating an account SAS

Allowed services
✓ Blob ☐ File ☐ Queue ☐ Table

Allowed resource types
☐ Service ☐ Container ✓ Object

Allowed permissions
☐ Read ✓ Write ☐ Delete ☐ List ✓ Add ✓ Create ☐ Update ☐ Process ☐ Immutable storage ☐ Permanent delete

Blob versioning permissions
✓ Enables deletion of versions

Allowed blob index permissions
✓ Read/Write ✓ Filter

Start and expiry date/time
Start 12/30/2025 12:54:23 PM
End 12/30/2025 9:09:23 PM
(UTC+02:00) --- Current Time Zone ---

Allowed IP addresses
For example, 168.1.5.65 or 168.1.5.65-168.1.5.70

Allowed protocols
[✓ HTTPS only] HTTPS and HTTP

Preferred routing tier
[✓ Basic (default)] Microsoft network routing Internet routing

Some routing options are disabled because the endpoints are not published.

Signing key
key1

Generate SAS and connection string

8. You will get connection string, SAS token and blob service SAS URL. Make sure you copy **only** the SAS token, for example: ?sv=2023-01-03&ss=b&srt=o&sp=acw&se=...

Security note about why this is secure:

- Token is write-only.
- Token is time-bound.
- Token is HTTPS-only.
- Token cannot list, read, or delete.
- Token cannot access other services.
- Token cannot enumerate containers.

Even if leaked:

- No data theft
- No data destruction
- Very limited blast radius

Still note that although standard users still cannot see it, SYSTEM and local admin could.

Option B — Stronger model (recommended for production)

Powershell upload script

Pre-requisites for the script

- container URL
- SAS token
- Powershell upload script (powerguardcloud_upload.ps1)
  - Make sure you change the following parameters in the script: $ContainerUrl and $SasToken.

- You can get the Container URL from Azure portal > storage accounts > YOUR STORAGE ACCOUNT (Powerguardlogs) > containers > powerguard-ps-transcripts > settings > properties > URL.

Intune deployment settings

- In intune admin center, navigate to Devices > Scripts and remediations > platform scripts > Add > Windows 10 and later.
- Give the powershell script a name, like: “powerguardcloud_upload”, then choose next.
- On script settings, upload the script and verify the following:
  - Run this script using the logged-on credentials: No
  - Run script in 64-bit PowerShell: Yes
  - Enforce script signature check: No
  - Assign to your single endpoint device group.
- On Assignments settings, assign the group contains the relevant endpoints, then choose next.
- On Review + Create choose Create
- To verify the scripts has been deployed, run the following powershell command:
  Get-ScheduledTask -TaskName "PowerGuard-Upload-PS-Transcripts"
  or
  Start-ScheduledTask -TaskName "PowerGuard-Upload-PS-Transcripts" to trigger the execution and then:
  Get-ChildItem C:\ProgramData\PowerGuard\Upload
  to verify the files are created

[Optional] – Microsoft Teams Workflow (webhook)

1. Go to the **Microsoft Teams** channel where you want alerts.
2. Click the **Three Dots (...)** next to the channel name > **Workflows**.
3. Search for **"Webhook"**.
4. Select the template: **"Post to a channel when a webhook request is received"**.
5. Name the flow (e.g., "PowerGuard Bot") and click **Next** / **Add workflow**.
6. **Copy the generated URL** (It will start with https://...logic.azure.com...).
7. Copy this URL for the PowerShell script that will be configured on Azure function section.

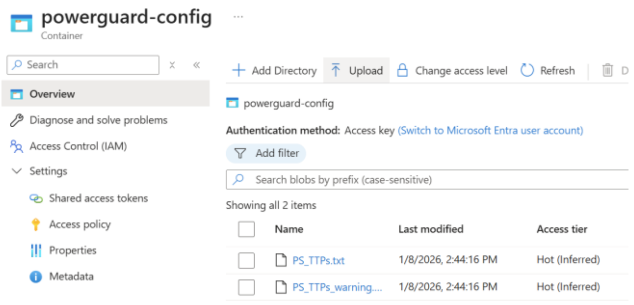
Azure function – investigate transcript content

We are deploying a serverless Azure Function (PowerShell Core) that automatically triggers whenever a new transcript lands in the powerguard-ps-transcript container.

- **Trigger:** New Blob in powerguard-ps-transcript
- **Configuration:** Reads threat intelligence from powerguard-config (PS_TTPs.txt & PS_TTPs_warning.txt)
- **Action:** Scans content, identifies matches, and sends categorized alerts (Critical vs. Warning).

Pre-requisites:

- Transcriptions are already uploaded to azure storage account's container.
- In case of Microsoft Teams integrations – you already created webhook URL for in Microsoft Teams workflow.
- run.ps1 script edited with the relevant webhook URL as value for $WebhookUrl.
- You have 2 lists of TTPs – standard and warning lists. The files' name should be:
    - PS_TTPs.txt (list of high severity TTPs)
    - PS_TTPs_warning.txt (list of warning severity TTPs since it might create high volume of false / positive)
- Upload these files to a new container in your storage account named **powerguard-config**. Don't put it in the transcript's container, or it will trigger the scanner loop!



powerguard-config

Authentication method: Access key (Switch to Microsoft Entra user account)

Showing all 2 items

Name	Last modified	Access tier
PS_TTPs.txt	1/8/2026, 2:44:16 PM	Hot (Inferred)
PS_TTPs_warning....	1/8/2026, 2:44:16 PM	Hot (Inferred)

Create the Function App

1. Go to the **Azure Portal** search for **Function App.**
2. Click **Create**
3. **On select a hosting option page, choose your best option – or choose "Consumption for" pay-as-you-go.**
4. Basics Tab:
    a. **Subscription**: Your Subscription.
    b. **Resource Group**: Select your existing resource group (e.g., PowerGuard-RG).
    c. **Function App Name**: PowerGuard-Scanner (or unique name).

   d. **Operating System**: Windows
   e. **Runtime stack**: PowerShell Core.
   f. **Version**: 7.4 (or latest recommended).
   g. **Region**: Same region as your Storage Account (e.g., West Europe).
5. Storage tab
   a. **Storage Account**: Select Create new.
   b. **Name**: powerguardconf
6. Click **Review + create** -> **Create**

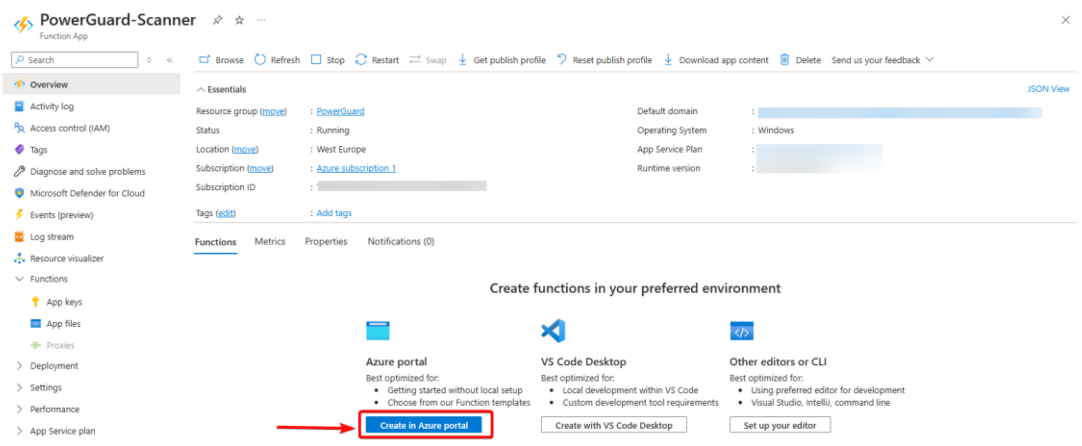
Connect the Data Storage

This part is relevant in case you create a new storage account to separate powershell transcription logs from powerguard system configurations. Since we created a separate system storage account, we must now explicitly connect the Function to your existing Data Storage where the logs are.

1. Go to your **Data Storage Account** (where powerguard-ps-transcript is).
2. On the left menu, click **Access keys.**
3. Click **Show** next to key1 and copy the **Connection string**.
4. Go to your new Function App (PowerGuard-Scanner).
5. On the left menu, under Settings, click Environment variables.
6. In the App settings tab, click + Add.
   - Name: TranscriptStorageConnection
   - Value: (Paste the connection string you copied)
7. Click Apply -> Apply (at the bottom) -> apply again in the App settings > Confirm.

Create the Trigger Function

1. In the Function App, scroll to **Functions** (bottom menu) and click **Create in Azure portal**.



Resource group (move) : PowerGuard
Status : Running
Location (move) : West Europe
Subscription (move) : Azure subscription 1
Subscription ID : [redacted]
Tags (edit) : Add tags
Default domain : [redacted]
Operating System : Windows
App Service Plan : [redacted]
Runtime version :

Create functions in your preferred environment

Azure portal
Best optimized for:
- Getting started without local setup
- Choose from our Function templates

VS Code Desktop
Best optimized for:
- Local development within VS Code
- Custom development tool requirements

Other editors or CLI
Best optimized for:
- Using preferred editor for development
- Visual Studio, IntelliJ, command line

2. Select **Azure Blob Storage trigger**.

3. **New Function Name:** ScanTranscript.

4. **Path:** powerguard-ps-transcript/{name}

5. **Storage account Connection:** Change this from AzureWebJobsStorage to TranscriptStorageConnection (the setting you just created).

6. Click **Create**.

Set Up Bindings (function.json)

1. Open your new ScanTranscript function.
2. On the left menu, click Code + Test.
3. Select function.json from the dropdown menu.
4. Replace the content with this JSON (Notice the connection property is now explicit):

```
{
  "bindings": [
    {
      "name": "InputBlob",
      "type": "blobTrigger",
      "direction": "in",
      "path": "powerguard-ps-transcript/{name}",
      "connection": "LogStorageConnection"
    },
    {
      "name": "CriticalKeywordsBlob",
      "type": "blob",
      "direction": "in",
      "path": "powerguard-config/PS_TTPs.txt",
      "connection": "LogStorageConnection"
    },
    {
      "name": "WarningKeywordsBlob",
      "type": "blob",
      "direction": "in",
      "path": "powerguard-config/PS_TTPs_warning.txt",
      "connection": "LogStorageConnection"
    }
  ]
}
```

Deploy the PowerShell Scanner Code

In the Function App, switch the file dropdown to run.ps1.

Troubleshooting

Issue: Powershell script is not deployed on endpoints

Solution 1: check for intune logs - C:\ProgramData\Microsoft\IntuneManagementExtension\Logs\IntuneManagementExtension.log
if an error like “Found 0 MDM certificates from Local Computer Store” appears but still see “Found **1** MDM certificates from Local User Store”, the assignment should be temporary for users until you will fix the intune certificate deplotment for local computer store.

Solution 2: Check for failed scheduled task or script deployment.

Select-String -Path "C:\ProgramData\Microsoft\IntuneManagementExtension\Logs\IntuneManagementE xtension.log" -Pattern "PowerShell script","Script execution","Download","Error","Failed","Succeeded" | Select-Object -Last 80

Issue: how to check powerguard upload log files

Run in Powershell:
Get-Content "C:\Windows\PowerGuard\Upload\upload.log" -Tail 80
it might give you a summary of the total amount of files uploaded (Uploaded=NUMBER) or failed to upload (Failed=NUMBER). If Failed contain numbers, it might be an authentication issue (SAS – Shared access signature - has expired) or network issue (the computer cannot access Azure network addresses)

In addition,

Get-content “c:\windows\powerguard\upload\azcopy-direct-DATE-TIME\LOGFILE.log file can tell you explicitly what the reason could be. For example:

AuthenticationErrorDetail: Signature not valid in the specified time frame: Start [Mon, 12 Jan 2026] - Expiry [Mon, 26 Jan 2026] - Current [Thu, 05 Feb 2026]

Meaning that the SAS has expired.