

# Lab #8: Setup of Tenant-Level Logging

## Purpose:

- We'll configure logging for Entra ID (aka Azure AD), which includes **Audit Log** and **Sign-in Log** activity. These logs will be sent to our Logs Analytics workspace.

**Note:** To recap, we've gradually configured logs to be ingested into our Logs Analytics workspace. In future labs, we'll configure logging for a few final resources (Activity Log, Key Vault, Blob Storage).

## Tasks:

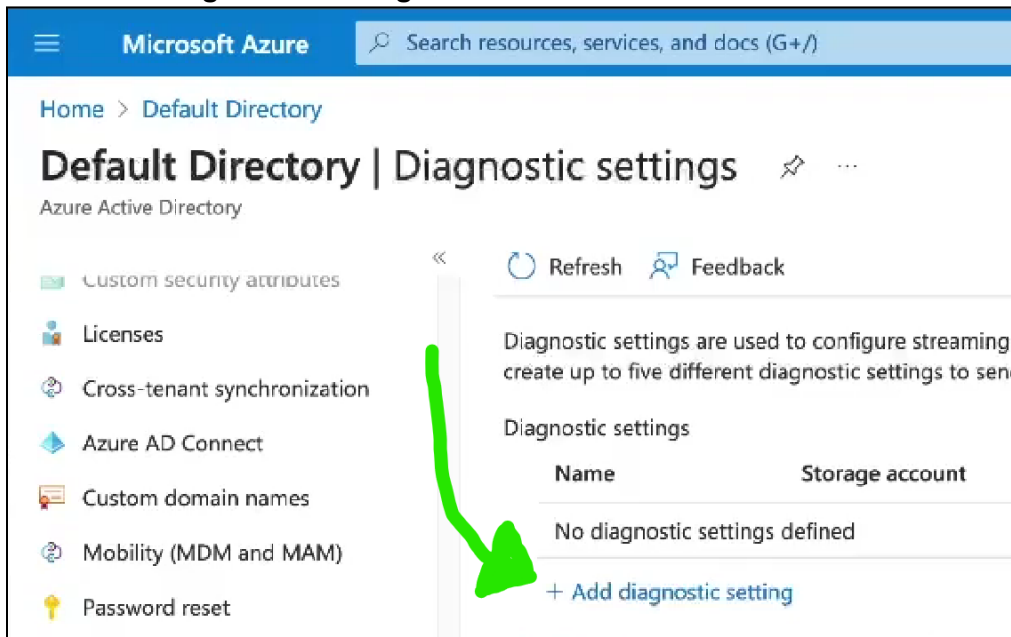
- Create diagnostic settings to ingest Entra ID logs**
- Generate audit logs (user creation, role assignment)**
  - Create a "Dummy" user account
  - Assign the role of Global Administrator
  - Delete the "Dummy" user account
- Observe the "Audit Logs" logs in Log Analytics Workspace**
  - Observe the logs for account creation, account deletion, and role assignment
  - Generate a specific log for "Global Administrator" role assignment
- Generate audit logs ("Failed Login" attempts)**
  - Create an "Attacker" user account
  - Generate logs for failed login attempts
  - Observe the newly created sign-in logs

## Task 1: Create diagnostic settings to ingest Entra ID logs

- Azure account > Entra ID > Diagnostic Settings >**

**Note:** We see many options for logs that we could ingest.

- Select **Add Diagnostic Setting >**

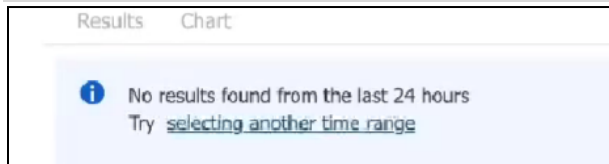


The screenshot shows the Microsoft Azure portal interface for configuring diagnostic settings for the Default Directory. The page title is "Default Directory | Diagnostic settings". On the left, there is a navigation menu with options like "Custom security attributes", "Licenses", "Cross-tenant synchronization", "Azure AD Connect", "Custom domain names", "Mobility (MDM and MAM)", and "Password reset". The main content area shows a table with columns "Name" and "Storage account". The table currently contains one row: "No diagnostic settings defined". Below the table is a "+ Add diagnostic setting" button. A green arrow points to this button.

- Name:** ds-audit-signin

- b. Logs > **Categories**: select the **AuditLogs** & **SigninLogs** checkboxes.
  - c. **Destination Details**: select the **Send to Log Analytics workspace** checkbox.
    - i. **Subscription**: (our Azure subscription)
    - ii. **Log Analytics workspace**: (our workspace)
  - d. Select **Save**.
3. Verify that the tables (which will hold the logs) have been created.
    - a. **Log Analytics workspace** > select our workspace > **Logs**
    - b. In **New Query** terminal, try each command:
      - i. **SigninLogs** > **Run**
      - ii. **AuditLogs** > **Run**

**Note:** Though no results have appeared yet, it confirms that the logging is working!



## Task 2: Generate some audit logs (user creation, role assignment)

**Note:** Each of these sub-tasks should generate separate audit logs.

### Create a “Dummy” user account:

1. **Azure** portal > **Entra ID** > **Users** > **Create New User** > fill out the fields:
  - a. **User principle name**: dummy\_user
  - b. **Mail nickname**: (select the **Derive from user principle name** checkbox)
  - c. **Display name**: dummy\_user
  - d. **Password**: (create a secure password)
  - e. **Account enabled**: (yes)
  - f. **User type**: Member
2. Log in once using dummy\_user’s credentials:
  - a. Private browser > portal.azure.com > Log in successfully using **dummy\_user**’s credentials.
  - b. Done. Close this browser.

### Assign the role of Global Administrator:

1. **Azure** portal > **Entra ID** > **Users** > select the **dummy\_user** account **Assigned roles** >
  - a. **Add assignment** > search for “Global Administrator” > **Add**.

### Delete the “Dummy” user account:

1. **Azure** portal > **Entra ID** > **Users** >
  - a. Select the checkbox next to the **dummy\_user** account > **Delete account**

## Task 3: Observe the “Audit Logs” logs in Log Analytics Workspace

### Observe the logs for account creation, account deletion, and role assignment:

**Note:** We’ll now inspect the logs that we’ve recently generated (in the last task).

1. **Azure** portal > **Log Analytics workspace** > select your workspace > **Logs**
  - a. In the query box, run **AuditLogs**
  - b. Locate the first “**Add user**” log, and then select “>” to expand the log.

**Note:** The screenshot below shows a log for user account generation.

1 AuditLogs

TimeGenerated [UTC] ↑↓	ResourceId	OperationName
> 9/8/2023, 3:51:29.683 PM	/tenants/0c731313-3d70-419b-b042-a6b46dde761f/pr...	<u>Add user</u>

TimeGenerated [Local Time] ↑↓	ResourceId
OperationName	<u>Add user</u>
OperationVersion	1.0
Category	<u>UserManagement</u>

c. We can tell who created the user account.

TimeGenerated [Local Time] ↑↓	ResourceId	OperationName
AdditionalDetails		{}
Id	Directory_8363573f-5533-4644-993a-6be32754c83e_6ROMA_1	
InitiatedBy	{"user":{"id":"224d608a-dbd6-4512-b1ee-19bce4f7d6df","dis	
user	{"id":"224d608a-dbd6-4512-b1ee-19bce4f7d6df","displayName":null,"userPrincipa	
displayName	null	
id	224d608a-dbd6-4512-b1ee-19bce4f7d6df	
ipAddress	2601:601:a300:ea10:8d7a:7d03:3fb3:3ecd	
roles	[]	
<u>userPrincipalName</u>	[REDACTED].com#EXT#@[REDACTED].onmicrosoft.com	

d. To show the most recent events, run [AuditLogs | order by TimeGenerated desc](#)

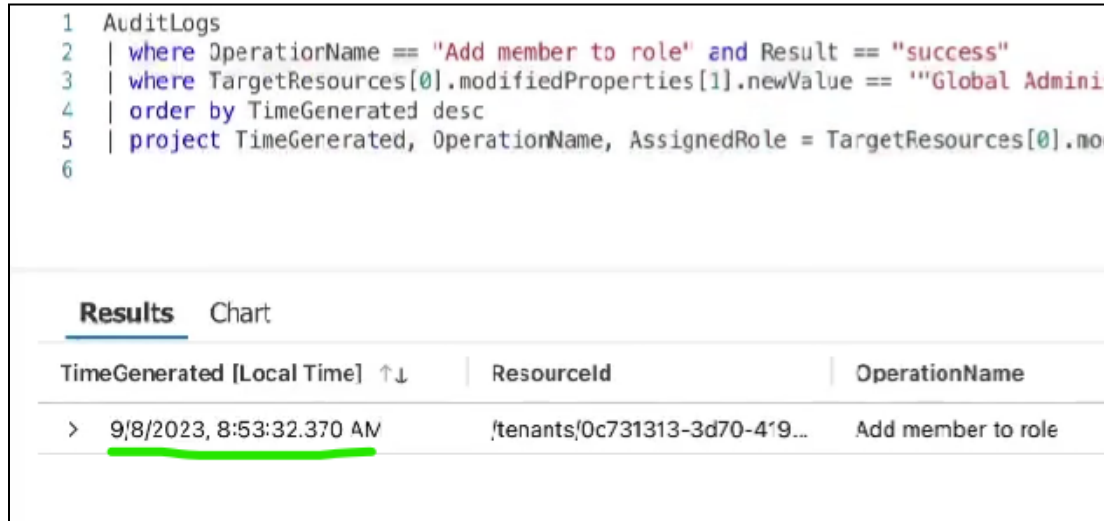
**Note:** Below we can see the generated logs for when we created a user, assigned a role, and then deleted the user account. You could expand these to inspect these logs further.

TimeGenerated [Local Time] ↑↓	ResourceId	OperationName
> 9/8/2023, 8:54:20.892 AM	/tenants/0c731313-3d70-419...	<u>Delete user</u>
> 9/8/2023, 8:53:32.370 AM	/tenants/0c731313-3d70-419...	<u>Add member to role</u>
> 9/8/2023, 8:53:12.747 AM	/tenants/0c731313-3d70-419...	Add service principal
> 9/8/2023, 8:52:14.519 AM	/tenants/0c731313-3d70-419...	Change password (self-service)
> 9/8/2023, 8:52:14.511 AM	/tenants/0c731313-3d70-419...	Update StsRefreshTokenValidFrom Timestamp
> 9/8/2023, 8:52:14.510 AM	/tenants/0c731313-3d70-419...	Change user password
> 9/8/2023, 8:52:13.720 AM	/tenants/0c731313-3d70-419...	Add service principal
> 9/8/2023, 8:51:29.683 AM	/tenants/0c731313-3d70-419...	<u>Add user</u>

## Generate a specific log for “Global Administrator” role assignment:

**Note:** Logs like this are beneficial because it could alert on unauthorized privilege escalation attempts.

1. **Azure** portal > **Log Analytics workspace** > select your workspace > **Logs**
  - a. In the query box, run this command:  
**AuditLogs | where OperationName == "Add member to role" and Result == "success" | where TargetResources[0].modifiedProperties[1].newValue == "Global Administrator" or TargetResources[0].modifiedProperties[1].newValue == "Company Administrator" | order by TimeGenerated desc | project TimeGenerated, OperationName, AssignedRole = TargetResources[0].modifiedProperties[1].newValue, Status = Result, TargetResources**



```
1 AuditLogs
2 | where OperationName == "Add member to role" and Result == "success"
3 | where TargetResources[0].modifiedProperties[1].newValue == "Global Administrator"
4 | order by TimeGenerated desc
5 | project TimeGenerated, OperationName, AssignedRole = TargetResources[0].modifiedProperties[1].newValue, Status = Result, TargetResources
```

**Results** Chart

TimeGenerated [Local Time] ↑↓	ResourceId	OperationName
> 9/8/2023, 8:53:32.370 AM	/tenants/0c731313-3d70-4f9...	Add member to role

**Note:** We see one log for when we assigned the Global Administrator role to the “Dummy” user.

## Task 4: Generate audit logs for “Failed Login” attempts

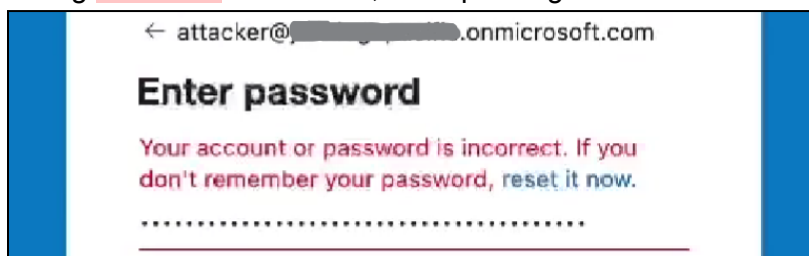
### Create an “Attacker” user account:

1. **Azure** portal > **Entra ID** > **Users** >
  - a. **Create New User** > fill out the fields:
    - i. User principle name: attacker
    - ii. Mail nickname: (select the **Derive from user principle name** checkbox)
    - iii. Display name: attacker
    - iv. Password: (create a secure password)
    - v. Account enabled: (yes)
    - vi. User type: Member
  - b. Select **Create**.

### Generate logs for failed login attempts:

**Note:** We want to generate several logs for “bad login” attempts.

1. Open an **incognito browser** > **Azure** portal.
  - a. Using **correct** credentials, sign into the “attacker” account > sign out.
  - b. Using **incorrect** credentials, attempt to sign into the “attacker” account > do this 10x.



- c. Try signing in using **correct** credentials (the account may be locked).

**Note:** It may take 30-60 min for the new sign-in logs to appear in your Log Analytics workspace.

## Observe the newly created sign-in logs:

1. Azure portal > Log Analytics workspace > select our workspace > Logs
  - a. Run the **SigninLogs** command in the query box. Expand the log results by selecting ">".

Results		Chart
TimeGenerated [Local Time] ↑↓	ResourceId	OperationName
9/8/2023, 10:48:14.871 AM	/tenants/0c731313-3d70-4...	Sign-in activity
> 9/8/2023, 10:48:02.483 AM	/tenants/0c731313-3d70-419...	Sign-in activity
> 9/8/2023, 10:48:02.483 AM	/tenants/0c731313-3d70-419...	Sign-in activity
> 9/8/2023, 10:47:36.529 AM	/tenants/0c731313-3d70-419...	Sign-in activity
> 9/8/2023, 10:47:34.607 AM	/tenants/0c731313-3d70-419...	Sign-in activity
> 9/8/2023, 10:46:54.650 AM	/tenants/0c731313-3d70-419...	Sign-in activity
> 9/8/2023, 10:46:32.652 AM	/tenants/0c731313-3d70-419...	Sign-in activity

- b. In the query box, run this extended command:

```
SigninLogs | where ResultDescription == "Invalid username or password or Invalid on-premise username or password." | extend location = parse_json(LocationDetails) | extend City = location.city, State = location.state, Country = location.countryOrRegion, Latitude = location.geoCoordinates.latitude, Longitude = location.geoCoordinates.longitude | project TimeGenerated, ResultDescription, UserPrincipalName, AppDisplayName, IPAddress, IPAddressFromResourceProvider, City, State, Country, Latitude, Longitude
```

**Note:** That extended query generates log results for failed login attempts, and only presents the variables/fields that we specified (e.g., IP address, location details, etc.).

```
1 SigninLogs
2 | where ResultDescription == "Invalid username or password or Invalid on-premise username or password."
3 | extend location = parse_json(LocationDetails)
4 | extend City = location.city, State = location.state, Country = location.countryOrRegion, Latitude = location.geoCoordinates.latitude, Longitude = location.geoCoordinates.longitude
5 | project TimeGenerated, ResultDescription, UserPrincipalName, AppDisplayName, IPAddress, IPAddressFromResourceProvider, City, State, Country, Latitude, Longitude
6
7
```

Results		Chart		
TimeGenerated [Local Time] ↑↓	ResultDescription	UserPrincipalName	AppDisplayName	IPAddress
> 9/8/2023, 10:48:02.483 AM	Invalid username or password or Invalid on-premise username or password.	attacker@...on...	Azure Portal	...
> 9/8/2023, 10:47:34.607 AM	Invalid username or password or Invalid on-premise username or password.	attacker@...on...	Azure Portal	...
> 9/8/2023, 10:46:31.366 AM	Invalid username or password or Invalid on-premise username or password.	attacker@...on...	Azure Portal	...
> 9/8/2023, 10:46:31.366 AM	Invalid username or password or Invalid on-premise username or password.	attacker@...on...	Azure Portal	...
> 9/8/2023, 10:46:31.366 AM	Invalid username or password or Invalid on-premise username or password.	attacker@...on...	Azure Portal	...

**Note:** The above screenshot shows an example of an expanded log entry and its parsed LocationDetails JSON object.

End:

- In the next lab, we'll be configuring the subscription-level logs (e.g., Activity Log).