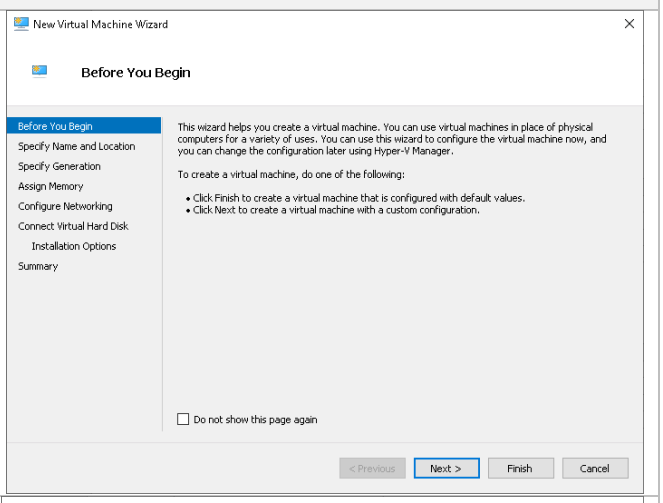
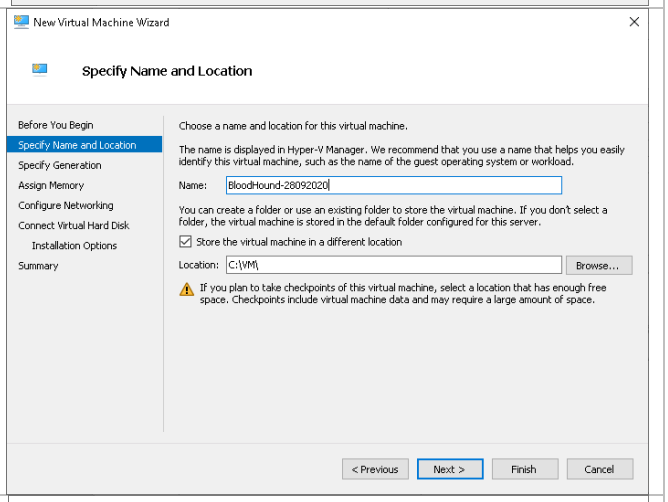
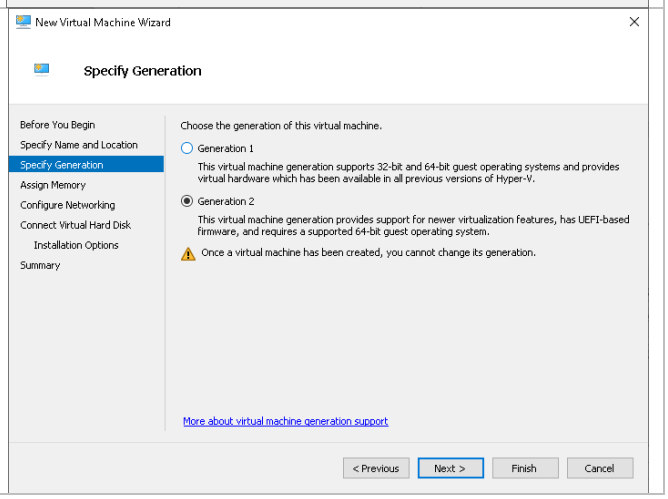
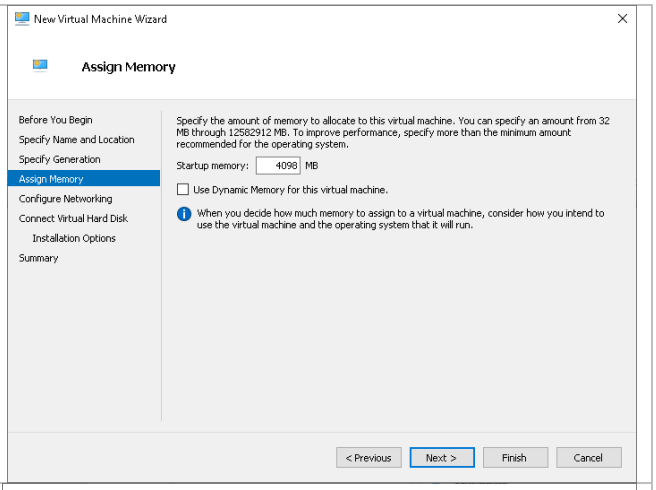
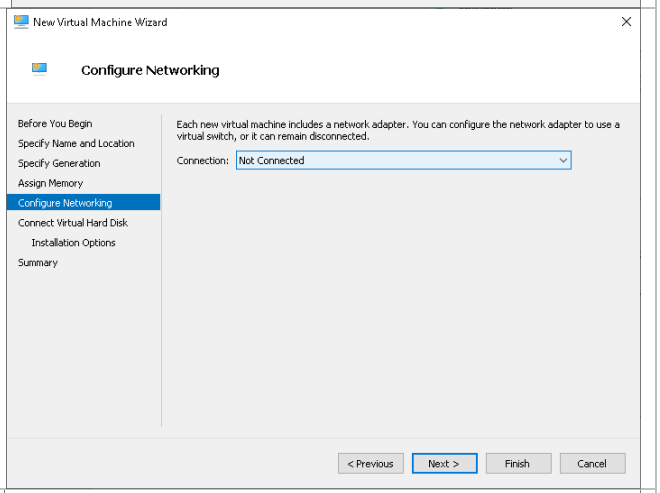
Assessment VM

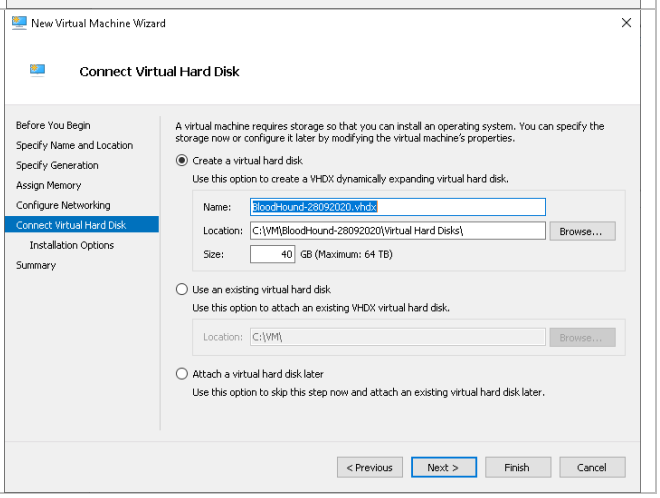
Create and Prepare a VM

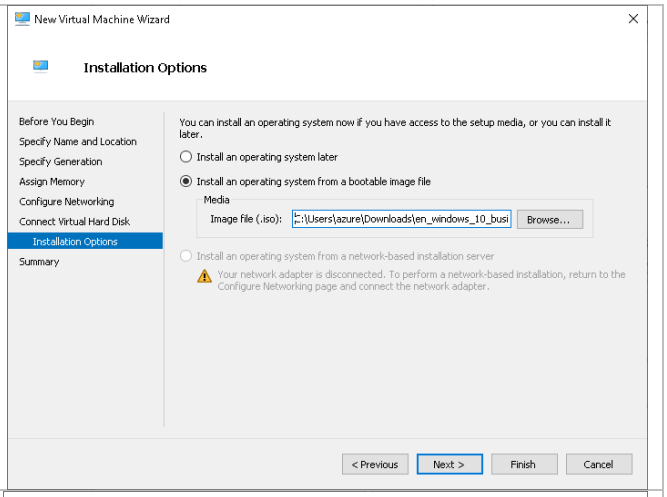
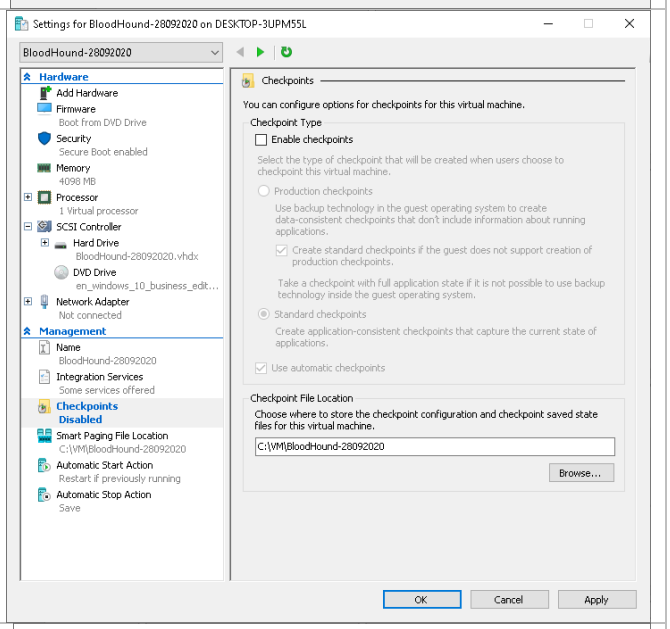
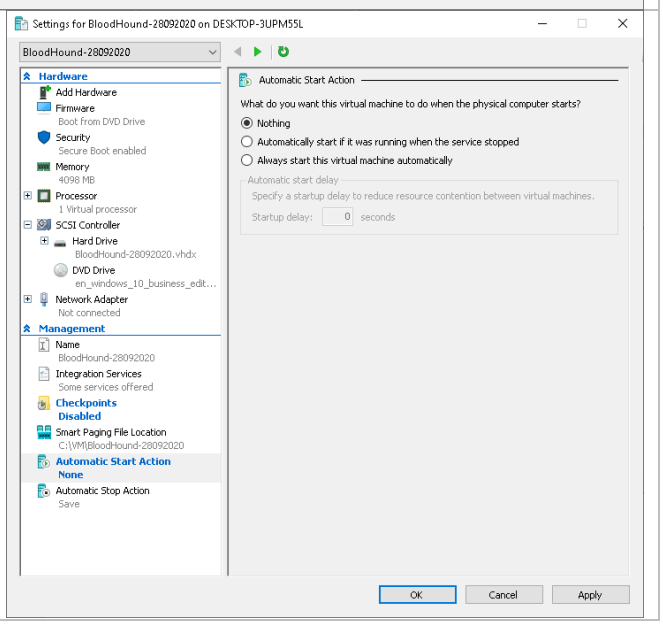
	
	
	

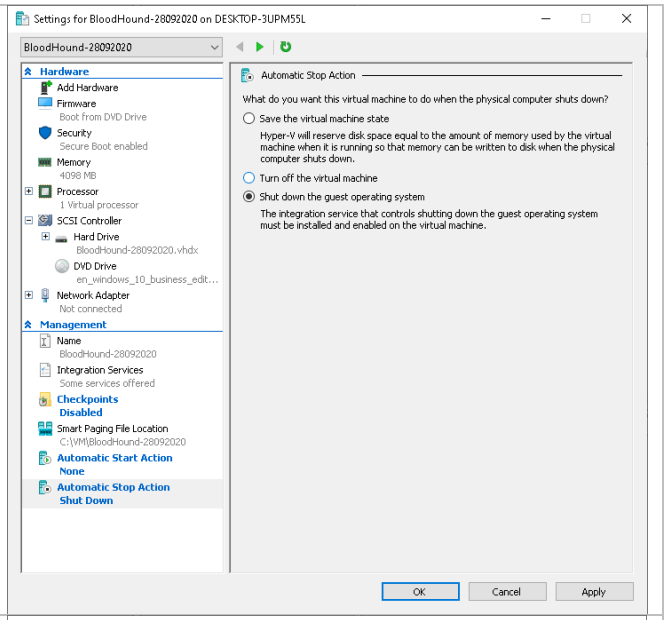
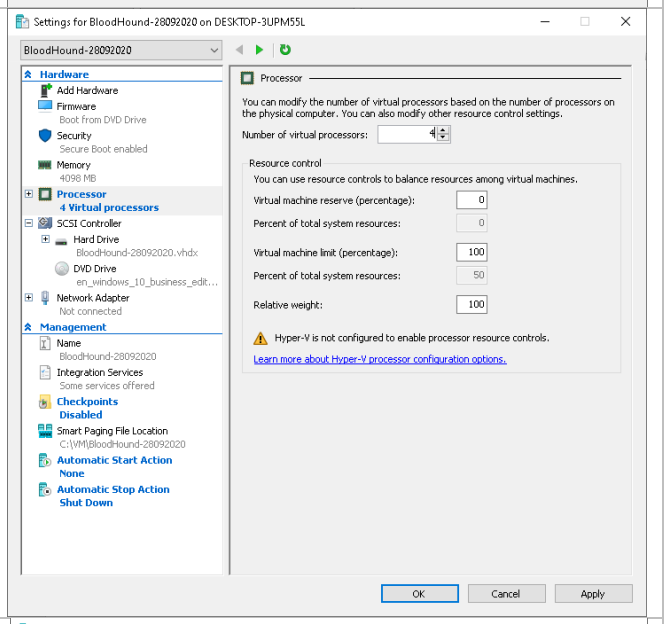
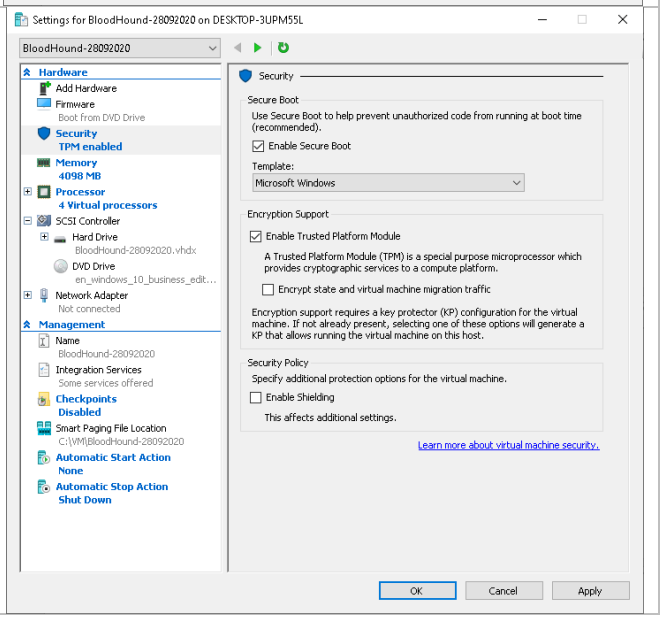


Chose “Not Connected” for the moment

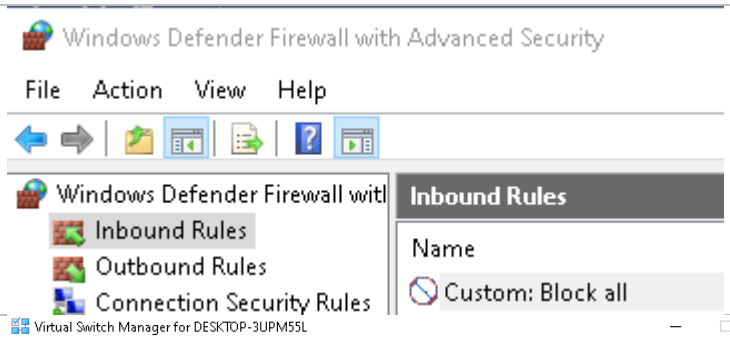
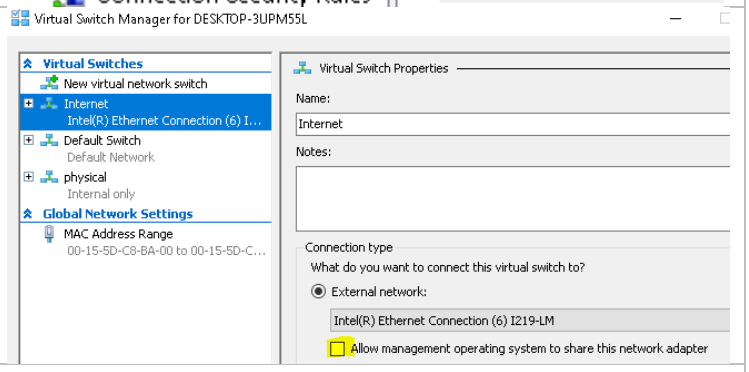




	
Checkpoints: Disable Checkpoints (Optional)	
Automatic Start Action: Turn off Automatically start	

Automatic Stop Action: Shut down the guest operating system	
Processor: Switch to 4 virtual CPU's (depends on the machine)	
Security: Enable Secure Boot and Enable TPM	

Install OS

Detach any Network Cable (No Internet Connection required at this stage)	
Install Windows OS (Most recent version)	
Create an inbound rule a block all the traffic	
If a VM is used, assign a dedicated VM Switch and don't share it with the host.	
Install Security Updates	Disconnect after update is complete. Exposure to the Network must be as minimal as possible.

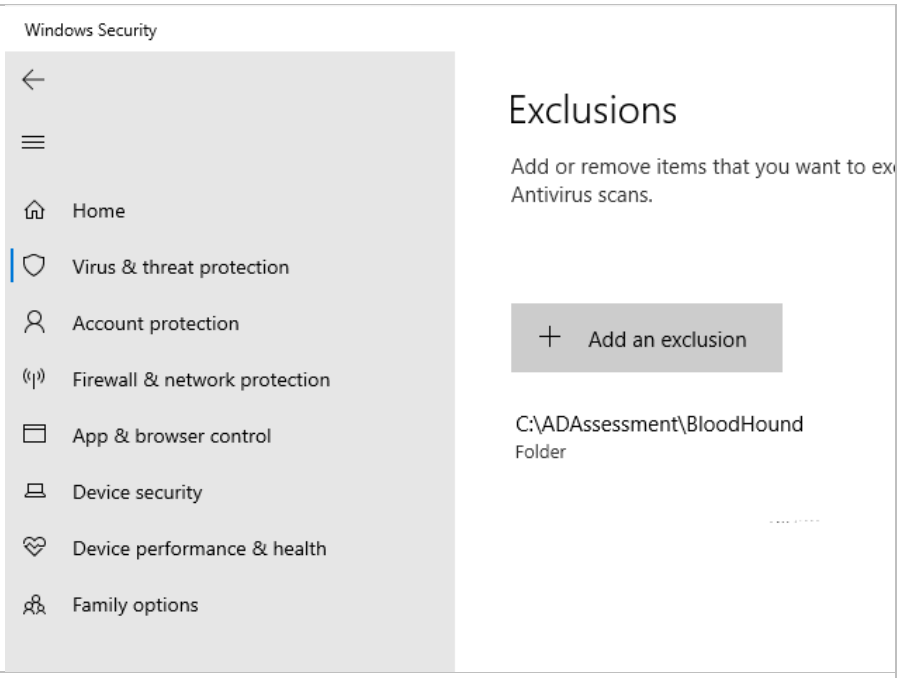
BloodHound (AD + Azure Assessment)

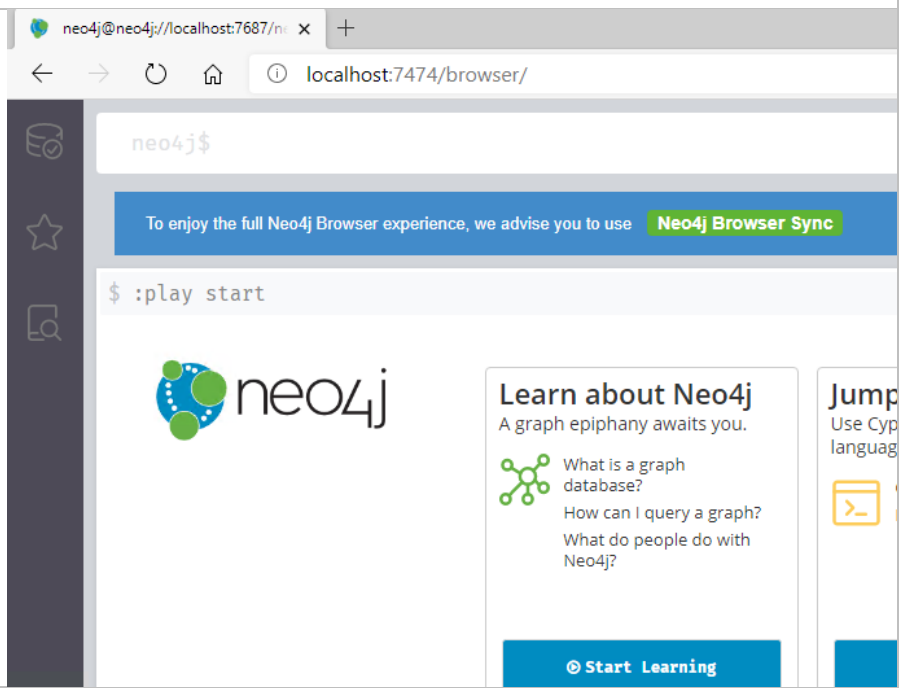
Prepare Assessment Client (Windows)

Either use a dedicated machine for the assessment or create a VM on an assessment machine.

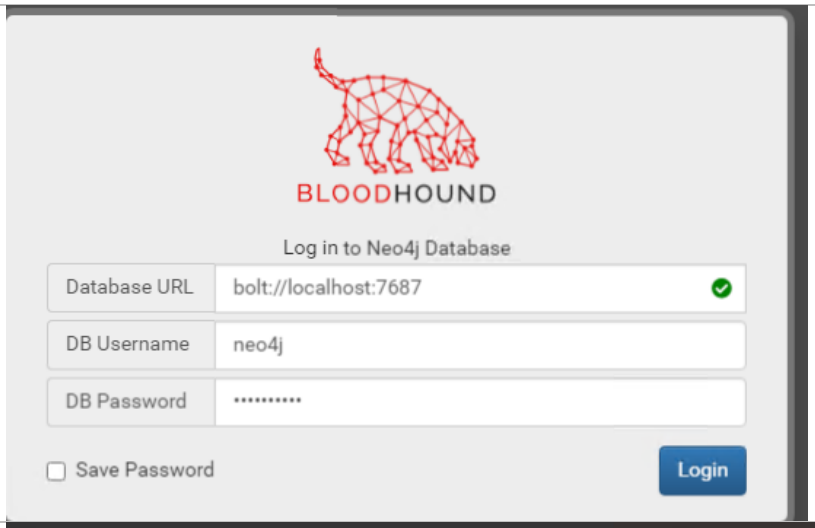
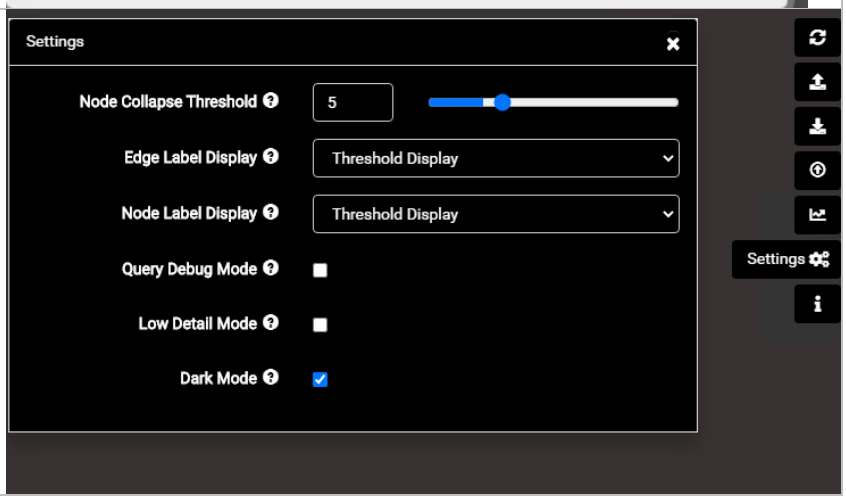
See First Chapter for VM preparation: **Error! Reference source not found.**

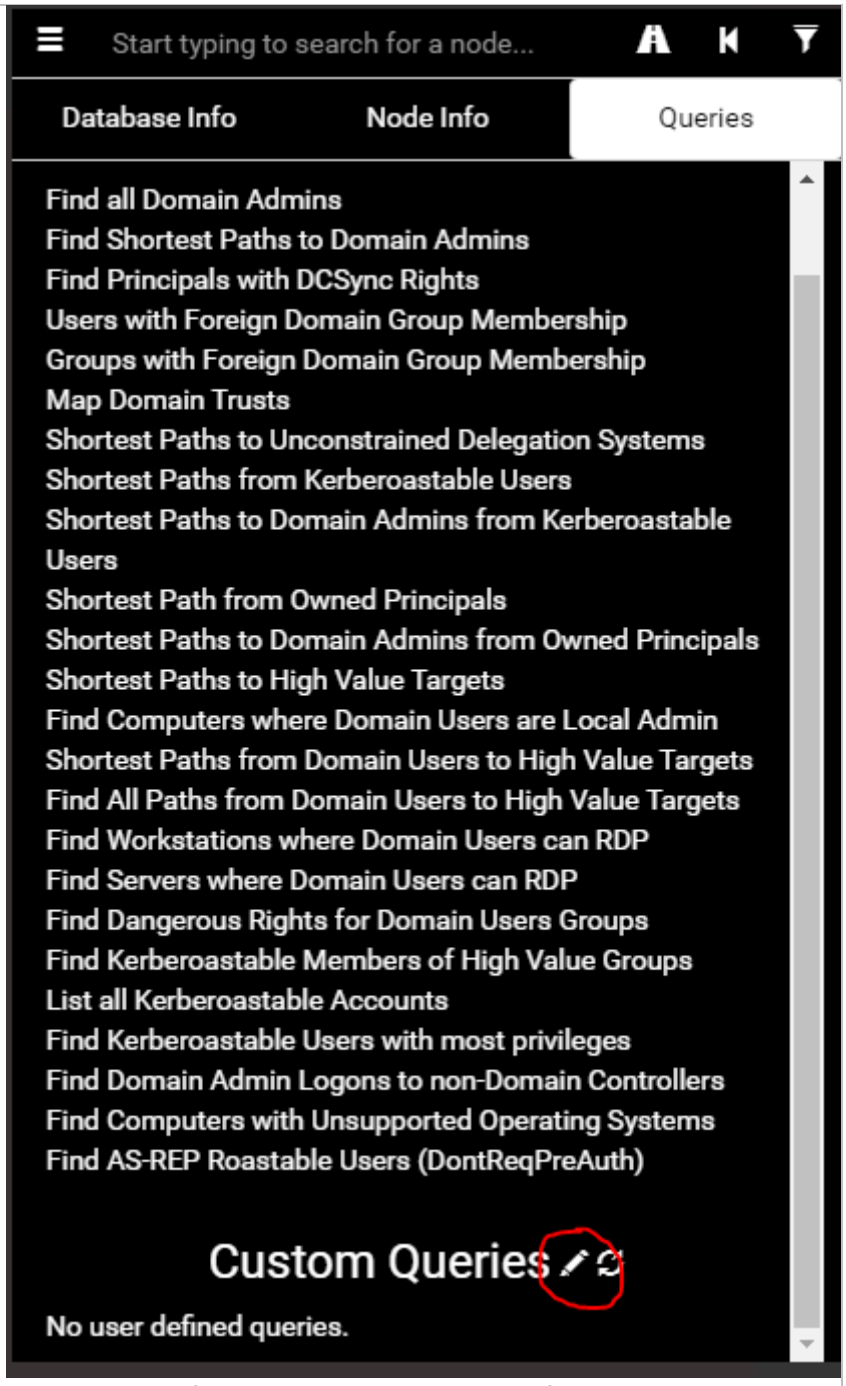
Create a C:\ADAssessment directory	
Create a C:\ADAssessment\BloodHound directory	

Create a Defender exclusion for the Folder. Virus & Threat protection settings > Exclusions: C:\ADAssessment\BloodHound	
Create folder: C:\ADAssessment\source You can place all the following source files into that folder	
Download Neo4j Community Edition database engine	https://neo4j.com/download-center/#community
Download the latest version of the BloodHound GUI + Source Code	Releases · BloodHoundAD/BloodHound (github.com)
Download CustomFilter	Bloodhound-Custom-Queries/customqueries.json at master · hausec/Bloodhound-Custom-Queries (github.com)
Download Zulu JDK 11	Java Download | Java 8, Java 11, Java 13 - Linux, Windows & macOS (azul.com) 
Install Zulu JDK	
Extract Bloodhound binaries to C:\ADAssessment\BloodHound	
Extract neo4j into the C:\ADAssessment\BloodHound directory	
Open cmd	
Change folder to:	

C:\ADAssessment\BloodHound directory\neo4j…	
Run: Neo4j.bat install-service net start neo4j	PS C:\ADAssessment\BloodHound\neo4j-community-4.0.8\bin> .\Neo4j.bat install-service Neo4j service installed PS C:\ADAssessment\BloodHound\neo4j-community-4.0.8\bin> net start neo4j The Neo4j Graph Database - neo4j service is starting. The Neo4j Graph Database - neo4j service was started successfully.
Open the administrative web interface in the browser by going to http://localhost:7474 ➔ Username: neo4j ➔ Password: neo4j	
Change Password to "Bloodhound"	

BloodHound – Configuration (Windows)

Start Bloodhound from C:\ADAssessment\BloodHound User: neo4j Pass: Bloodhound	
Switch to Dark Mode	

Load CustomFilters	 Copy content of downloaded customqueries file to opened editor Or copy file to: C:\Users\<user>\AppData\Roaming\bloodhound

AD: SharpHound – Run (Windows)

AD Pre-requisites

Create a temporary assessment user in AD	

User Right: Domain User	
SAM-R: If possible assign temporary rights to the user to read SAM-R from all available Clients in the network.	

Run SharpHound to collect data

Open CMD	
`cd C:\ADAssessment\Bloodhound\resources\app\Collectors`	

`SharpHound.exe --domain <domain name>`
 	
If the assessment client is not domain joined:
 	
`runas /user:<domain>\adassessment /netonly cmd`	

Run SharpHound to collect Session data

Open CMD	
cd \Ingestors	
SharpHound.exe --domain <domain name> --CollectionMethod Session --Loop --Loopduration 03:00:00	3h Loop to collect only session data
Before loading the data decompress the main zip file (e.g. 20201014101654_BloodHoundLoopResults.zip) to get the result zip files. Import of the main zip file will not work.	

Azure: AzureHound

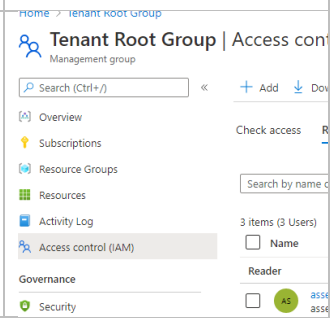
Pre-qreuisites

https://bloodhound.readthedocs.io/en/latest/index.html#collect-your-first-dataset

Open Powershell as Administrator	

Run:
 	
`[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12`
 	
`Set-ExecutionPolicy bypass`	

Install Azure CLI `Install-Module -Name Az -Scope CurrentUser -Repository PSGallery -Force`	
Install AzureAD Powershell Module `Install-Module AzureAD -Scope CurrentUser -Repository PSGallery -Force`	

Import AzureHound Modules `Import-Module C:\ADAssessment\Bloodhound\resources\app\Collectors\AzureHound.ps1`	
Create a temporary assessment user in Azure AD	
Assign the Azure AD Role via PIM or permanent: Global Reader	
Assign the Reader Azure Role via PIM to the Tenant Root group	

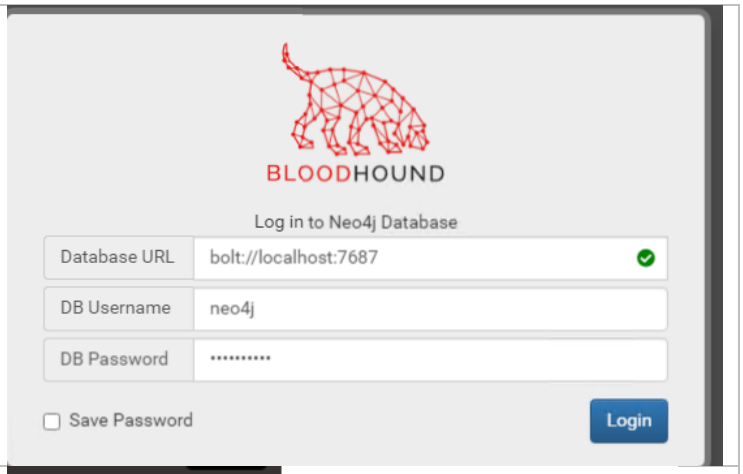
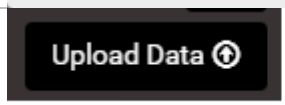
AzureHound – Run (Windows)

Open Powershell as Administrator	
login to Azure PowerShell `Connect-AzAccount`	
Login zu Azure AD `Connect-AzureAD`	

OPTIONAL:
 	
It is also possible to steal the access tokens from a compromised machine if that machine has been used to login to Azure PowerShell before. Copy the existing files:
 	
`C:\users\[Username]\.azure\AzureRmContextSettings.json`
`C:\users\[Username]\.azure\TokenCache.dat`
 	
And place them in your own .azure folder. Re-launch PowerShell and the token will now be used.	

Run `Invoke-AzureHound -TenantId <TenantID> -OutputDirectory C:\ADAssessment\Bloodhound\resources\app\Collectors`	

Load Data (Windows)

Start Bloodhound from C:\ ADAssessment\BloodHound User: neo4j Pass: Bloodhound	
Once Bloodhound has logged in, you will have a huge blank window. We need to load data into this. Click Upload Data Select the zip file collected by Sharphound or AzureHound. <datatime>_azurecollection.zip <datatime>_BloodHound.zip	
Wait till the data is imported	
Mark all T0 services as High Value Target: Right click -> High Value	AADConnect, ADFS, etc.

View Graph

Open: C:\ADAssessment\Bloodhound\BloodHound.exe	

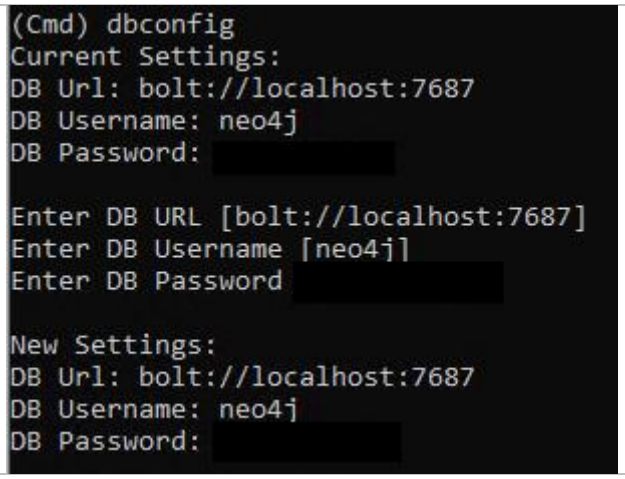
Create Excel Report (Windows)

Pre-requisites

Download Python (https://www.python.org/downloads)	

Install Python	
Open CMD and install the following modules: pip install neo4j pip install openpyxl	
Download BloodHound Analytics Python file from: [Bloodhound/bloodhoundanalytics.py at main · m8r1us/Bloodhound · GitHub] Save to: C:\ADAssessment\Report	Script made for neo4j >=V4.0

Create Report

Run: python bloodhoundanalytics.py <domain>	
Type: dbconfig Check the connection settings	
Type: Connect	

Type: startanalysis	
Excel is required to open the file	

Create Jupyter Notebook Report

Jupyter: [Installation — JupyterLab 3.0.16 documentation]

Plots: [plotly/plotly.py: The interactive graphing library for Python (includes Plotly Express) (github.com)]

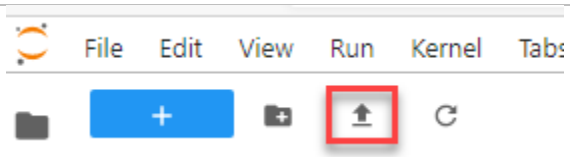
Pre-requisites

Download Python (if not already done) (https://www.python.org/downloads)	
Install NPM: [Node.js (nodejs.org)]	

Open cmd as admin and run:
 	
`pip install jupyterlab`
`pip install py2neo`
`pip install altair`
`pip install pandas`
`pip install psutil`
`npm install --save plotlywidget`
`jupyter labextension install jupyterlab-plotly@4.14.3`	

Create folder C:\ADAssessment\Reports	
Download bloodhound_report.ipynb from: https://github.com/m8r1us/Bloodhound/blob/main/bloodhound_report.ipynb and save the file to the Reports folder.	

Open Report

Open cmd as admin and run: `jupyter-lab`	
A browser should have opened automatically otherwise go to: http://localhost:8888/lab/workspaces	
Click on the upload file symbol and upload the "bloodhound_report.ipynb"	
Double-click or right click to open the bloodhound_report.ipynb file	

Change the connection string in step 2 accordingly.	Initialize BloodHound neo4j Database Connection [2]: g = Graph("bolt://localhost:7687", auth=("neo4j", "Bloodhound")) g [2]: Graph Neo4j connection URL See: http://localhost:7474/browser/ DB user Password
Change the Domain to the domain to assess	Domain to assess [3]: domain = "ASSESSMENT.INTRA"
Press the restart kernel button	Restart Kernel? Do you want to restart the current kernel? All variables will be lost. Cancel Restart

Cypher Queries (Azure)

Return All Azure Users that are part of the 'Global Administrator' Role	MATCH p =(n)-[r:AZGlobalAdmin*1..]->(m) RETURN p
Return All On-Prem users with edges to Azure	MATCH p=(m:User)-[r:AZResetPassword|AZOwns|AZUserAccessAdministrator|AZContributor|AZAddMembers|AZGlobalAdmin|AZVMContributor|AZOwnsAZAvereContributor]->(n) WHERE m.objectid CONTAINS 'S-1-5-21' RETURN p
Find all paths to an Azure VM	MATCH p = (n)-[r]->(g:AZVM) RETURN p
Find all paths to an Azure KeyVault	MATCH p = (n)-[r]->(g:AZKeyVault) RETURN p
Return All Azure Users and their Groups	MATCH p=(m:AZUser)-[r:MemberOf]->(n) WHERE NOT m.objectid CONTAINS 'S-1-5' RETURN p
Return All Azure AD Groups that are synchronized with On-Premise AD	MATCH (n:Group) WHERE n.objectid CONTAINS 'S-1-5' AND n.azsyncid IS NOT NULL RETURN n
Find all Privileged Service Principals	MATCH p = (g:AZServicePrincipal)-[r]->(n) RETURN p
Find all Owners of Azure Applications	MATCH p = (n)-[r:AZOwns]->(g:AZApp) RETURN p
Return All Azure Users (Console)	MATCH (n:AZUser) return n.azname
Return All Azure Applications	MATCH (n:AZApp) return n.objectid
Return All Azure Devices	MATCH (n:AZDevice) return n.name
Return All Azure Groups	MATCH (n:AZGroup) return n.name
Return all Azure Key Vaults	MATCH (n:AZKeyVault) return n.name

Return all Azure Resource Groups	MATCH (n:AZResourceGroup) return n.name
Return all Azure Service Principals	MATCH (n:AZServicePrincipal) return n.objectid
Return all Azure Virtual Machines	MATCH (n:AZVM) return n.name
Find All Principals with the 'Contributor' role	MATCH p = (n)-[r:AZContributor]->(g) RETURN p

ROADTools (Azure Assessment)

[dirkjanm/ROADtools: The Azure AD exploration framework. (github.com)]

AzureAD / Azure Pre-requisites

Create a **temporary assessment user in Azure AD**	
Assign the Azure AD Role via PIM: Global Reader	
Assign the Reader Azure Role via PIM for Azure.	

Prepare Assessment Client (Windows)

Either use a dedicated machine for the assessment or create a VM on an assessment machine.

Create a folder: C:\AzureAssessment	
Create a folder: C:\AzureAssessment\roadtools	
Create a folder: C:\AzureAssessment\sources You can place all the following source files into that folder	
Download Python (https://www.python.org/downloads)	
Install Python	
Install Microsoft C++ Build Tools https://visualstudio.microsoft.com/thank-you-downloading-visual-studio/?sku=BuildTools&rel=16	
Download Roadtools from: Pipelines - Run 20210527.1 artifacts (azure.com) Or: dirkjanm/ROADtools: The Azure AD exploration framework. (github.com)	

Extract ROADtools.zip to: C:\AzureAssessment\roadtools\roadlib C:\AzureAssessment\roadtools\roadrecon	
Open cmd	
Run: Cd C:\AzureAssessment\roadtools pip install pipenv pipenv install roadlib/ pipenv install roadrecon/	

Run RoadRecon (Windows)

Open cmd Run: Cd C:\AzureAssessment\roadtools pipenv shell	
Use the created Azure AD Account Run: Roadrecon auth --device-code	
Run: Roadrecon gather	
Create Conditional Access Rule dump Run: Roadrecon plugin policies	

View Data with RoadRecon UI

Open cmd		
Cd C:\AzureAssessment\roadtools pipenv shell		
Roadrecon-gui		
Open Browser		

Export Data to BloodHound

Use the new Bloodhound Version with integrated Azure AD support.

~~Download the following repository~~ ~~https://github.com/dirkjanm/Bloodhound-AzureAD~~	
~~Extract to AzureAssessment\~~	

~~Download and install neo4j Community Edition (Follow installation guide from Bloodhound)~~	
~~Open Cmd~~ ~~Cd C:\AzureAssessment\roadtools~~ ~~Pipenv shell~~ ~~Roadrecon plugin bloodhound~~	
~~Download NodeJS/NPM (https://www.npmjs.com/get-npm)~~	
~~Open Cmd~~ ~~cd AzureAssessment\BloodHound-AzureAD-master~~ ~~NPM inall~~ ~~NPM run dev~~ ~~The application could be also compiled to an exe.~~	
~~Open the URL.~~ ~~Control +R if blank screen for refresh~~	
~~Import SharpHound Data~~	

Stormspotter (Azure Assessment)

https://github.com/Azure/Stormspotter

AzureAD / Azure Pre-requisites

Create a **temporary assessment user in Azure AD**	
Assign the Azure AD Role via PIM: Global Reader	
Assign the Reader Azure Role via PIM for Azure.	

Prepare Assessment Client (Windows - Docker)

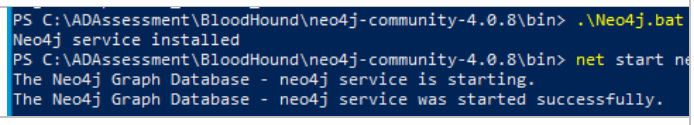
Either use a dedicated machine for the assessment or create a VM on an assessment machine. Docker will maybe not run on a VM.

https://github.com/Azure/Stormspotter#with-docker

Download and Install Docker (Follow the instruction to Install WSL2) Docker Desktop for Mac and Windows | Docker	
git clone https://github.com/Azure/Stormspotter	
Adjust ports etc. in the docker-compose.yaml if required. (Conflict with installed neo4j version)	
docker-compose up	

Prepare Assessment Client (Windows – Without Docker)

Either use a dedicated machine for the assessment or create a VM on an assessment machine.

Create a folder: C:\AzureAssessment	
Create a folder: C:\AzureAssessment\stormspotter	
Create folder: C:\AzureAssessment\source You can place all the following source files into that folder	
Download Python (https://www.python.org/downloads)	
Install Python 3.8.0 (https://www.python.org/ftp/python/3.8.0/python-3.8.0-amd64.exe)	
Download NodeJS/NPM (node-v14.17.0-x64) (https://www.npmjs.com/get-npm)	
Install NPM (NodeJS)	
Download Zulu JDK 11 (https://www.azul.com/downloads/zulu-community/?architecture=x86-64-bit&package=jdk)	
Install Zulu JDK	
Download Neo4j (https://neo4j.com/download-center/#community)	
Extract neo4j into the C:\AzureAssessment\Stormspotter directory	
Open cmd	
Change folder to: C:\AzureAssessment\Stormspotter\neo4j-community-4.2.6\bin	
Run: Neo4j.bat install-service net start neo4j	
Open the administrative web interface in the browser by going to	

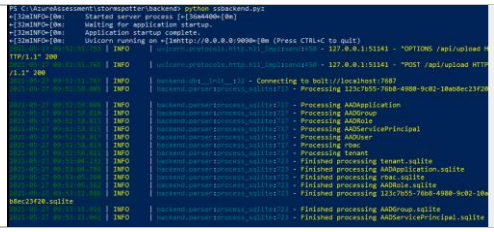
http://localhost:7474 Username: neo4j Password: neo4j	
Change Password to “stormspotter”	
Download Stormspotter (Releases · Azure/Stormspotter (github.com)	
Extract C:\AzureAssessment\stormspotter	
Install az cli powershell (https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-windows?tabs=azure-cli)	
Install Fronted reuirements Run: cd C:\AzureAssessment\stormspotter\frontend\dist\spa npm install -g @quasar/cli	

Run Stormcollector

Open separate CMD and RUN:
 	
`cd C:\AzureAssessment\Stormspotter\stormcollector`
 	
Run to show the help menu:
`python sscollector.pyz -h`
 	
Common options for all authentication types
 	
`python sscollector.pyz cli`
`python sscollector.pyz spn -t <tenant> -c <clientID> -s <clientSecret>`
 	
`--cloud: Specify a different Azure Cloud (GERMAN, CHINA, USGOV)`
`--config: Specify a custom configuration for cloud environments`
`--azure: Only enumerate Azure Resource Manager resources`
`--aad: Only enumerate Azure Active Directory`
`--subs: Subscriptions you wish to scan. Multiple subscriptions can be added as a space deliminated list.`
`--nosubs: Subscriptions you wish to exclude. Multiple subscriptions can be excluded as a space deliminated list.`
`--json: Convert SQLite output to JSON (WARNING: STORMSPOTTER ONLY PARSES SQLITE FORMAT )`
`This option is useful if you want to parse the output for reasons other than Stormspotter.`
`--ssl-cert: Specify an SSL cert for Stormcollector to use for requests. Not a common option`
`--backfill: Perform AAD enumeration only for object IDs associated with RBAC enumeration. Only applicable when --azure is specified.`	

Run to collect data by using the created azure assessment account:
 	
`Az login`
`python sscollector.pyz cli`	

Load Data (Windows)

Start Frontend -> Open CMD Run: cd C:\AzureAssessment\stormspotter\frontend\dist\spa quasar serve -p 9091 --history	
Start Backend -> open CMD (Required for uploading data) Run: cd C:\AzureAssessment\stormspotter\backend python ssbackend.pyz	
Open: http://localhost:9091	
Upload to Stormcollector Upload -> results_<date>.zip Files collected are in the folder: C:\AzureAssessment\stormspotter\stormcollector*.zip	
Check upload status in the Backend CMD Window	

Review Graph

Start Frontend -> Open CMD Run: cd C:\AzureAssessment\stormspotter\frontend\dist\spa quasar serve -p 9091 --history	
Open in Edge http://localhost:9091	

Cypher Queries

Show ServicePrincipal Relationships	MATCH (a)-[r]-(t) Where a.type ="AADServicePrincipal" RETURN *
Show all Global Administrators	MATCH (a:AADRole)<-[r:MemberOf]-(t) WHERE a.name = 'Global Administrator' RETURN *
Show all AAD Roles	MATCH (a:AADRole) RETURN *
Show full Tenant Relationships aka Christmastree	MATCH (a)-[r]-(t) Return *

AzureADAssessment

[GitHub - AzureAD/AzureADAssessment: Tooling for assessing an Azure AD tenant state and configuration]

Prepare Assessment Client

Create a folder: C:\AzureAssessment	
Create a folder: C:\AzureAssessment\AzureADAssessment	

Open Powershell and run:
 	
`Install-module msal.ps`
`Install-Module AzureADAssessment -Force`
 	
! If there are msal.ps install errors follow the on-screen recommendations and try again to install msal.ps before installing the AzureADAssessment module.	

## If you have already installed the module, run the following instead to ensure you have the latest version. `Update-Module AzureADAssessment -Force`	
Install PowerBi Download Microsoft Power BI Desktop from Official Microsoft Download Center	

Run AzureADAssessment

Use the created Azure AD Assessment Account
cd C:\AzureAssessment\AzureADAssessment
 	
`Connect-AADAssessment`
`Invoke-AADAssessmentDataCollection "C:\AzureAssessment\AzureADAssessment"`	

Create PowerBI Report	

`Complete-AADAssessmentReports AzureADAssessmentData-<TenantName>.onmicrosoft.com.zip -OutputDirectory "C:\AzureAssessment\AzureADAssessment"` Open PowerBi Template AzureADAssessment.pbit	
In the popup provide the path to the Results folder: `C:\AzureAssessment\AzureADAssessment\AzureADAssessmentData-<tenant>.onmicrosoft.com\AAD-<tenant>.onmicrosoft.com`	

Run AzureADAssessment on Hybrid Components

Export Portable Module `Export-AADAssessmentPortableModule "C:\AzureAssessment\AzureADAssessment"`	

Import the module on each server running hybrid components.
 	
`Import-Module "C:\AzureADAssessment\AzureADAssessmentPortable.psm1"`
 	
Export Data into a single output package.
 	
`Invoke-AADAssessmentHybridDataCollection "C:\AzureAssessment\AzureADAssessment"`	