GitHub - mandiant/Mandiant-Azure-AD-Investigator

By Willi Ballenthin (Google)

Archived: 2026-04-05 14:15:12 UTC

Focusing on UNC2452 TTPs

Overview

This repository contains a PowerShell module for detecting artifacts that may be indicators of UNC2452 and other threat actor activity. Some indicators are "high-fidelity" indicators of compromise, while other artifacts are so called "dual-use" artifacts. Dual-use artifacts may be related to threat actor activity, but also may be related to legitimate functionality. Analysis and verification will be required for these. For a detailed description of the techniques used by UNC2452 see our blog.

This tool is **read-only**. It does not make any changes to the Microsoft 365 environment.

In summary this module will:

- Do a best effort job at identifying indicators of compromise that will require further verification and analysis

It will *not*:

- Identify a compromise 100% of the time, or
- Tell you if an artifact is legitimate admin activity or threat actor activity.

With community feedback, the tool may become more thorough in its detection of IOCs. Please open an issue, submit a PR, or contact the authors if you have problems, ideas, or feedback.

Features

Federated Domains (Invoke-MandiantAuditAzureADDomains)

This module uses MS Online PowerShell to look for and audit federated domains in Azure AD. All federated domains will be output to the file `federated domains.csv`.

- **Signing Certificate Unusual Validity Period** - Alerts on a federated domain where the signing certificates have a validity period of > 1 year. AD FS managed certificates are valid for only one year. Validity periods that are longer than one year could be an indication that a threat actor has tampered with the domain federation settings. They may also be indicative of the use of a legitimate custom token-signing certificate. Have your administrators verify if this is the case.
- **Signing Certificate Mismatch** - Alerts on federated domains where the issuer or subject of the signing certificates do not match. In most cases the token-signing certificates will always be from the same issuer

and have the same subject. If there is a mismatch, then it could be an indication that a threat actor has tampered with the domain federation settings. Have your administrators verify if the subject and issuer names are expected, and if not consider performing a forensic investigation to determine how the changes were made and to identify any other evidence of compromise.

- **Azure AD Backdoor (any.sts)** - Alerts on federated domains configured with `any.sts` as the Issuer URI. This is indicative of usage of the Azure AD Backdoor tool. Consider performing a forensic investigation to determine how the changes were made and to identify any other evidence of compromise.
- **Federated Domains** - Lists all federated domains and the token issuer URI. Verify that the domain should be federated and that the issuer URI is expected.
- **Unverified Domains** - Lists all unverified domains in Azure AD. Unverified domains should not be kept in Azure AD for long in an unverified state. Consider removing them.

Examples

```
 !! Evidence of AAD backdoor found.
Consider performing a detailed forensic investigation
Domain name: foobar.com
Domain federation name:
Federation issuer URI: http://any.sts/16B45E3B
```

‼️ The script has identified a domain that has been federated with an issuer URI that is an indicator of an Azure AD Backdoor. The backdoor sets the issuer URI to hxxp://any.sts by default. Consider performing a forensic investigation to determine how the changes were made and identify any other evidence of compromise.

```
 !! A token signing certificate has a validity period of more than 365 days.
This may be evidence of a signing certificate not generated by AD FS.
Domain name: foobar.com
Federation issuer uri: http://sts.foobar.com
Signing cert not valid before: 1/1/2020 00:00:00
Signing cert not valid after: 12/31/2025 23:59:59
```

⚠️ The script has identified a federated domain with a token-signing certificate that is valid for longer than the standard 365 days. Consult with your administrators to see if the token-signing certificate is manually managed and if it is expected to have the stated validity period. Consider performing a forensic investigation if this is not expected.

Service Principals (Invoke-MandiantAuditAzureADServicePrincipals)

This module uses Azure AD PowerShell to look for and audit Service Principals in Azure AD.

- **First-party Service Principals with added credentials** - First-party (Microsoft published) Service Principals should not have added credentials except in rare circumstances. Environments that are or were previously in a hybrid-mode may have credentials added to Exchange Online, Skype for Business, and

AAD Password Protection Proxy Service Principals. Verify that the Service Principal credential is part of a legitimate use case. Consider performing a forensic investigation if the credential is not legitimate.

- **Service Principals with high level privileges and added credentials** - Identifies Service Principals that have high-risk API permissions assigned and added credentials. While the Service Principal and added permissions are likely legitimate, the added credentials may not be. Verify that the Service Principal credentials are part of a legitimate use case. Verify that the Service Principal needs the listed permissions.

Examples

```
 !! Identified first-party (Microsoft published) Service Principals with added credentials.
Only in rare cases should a first-party Service Principal have an added credential.
Verify that the added credential has a legitimate use case and consider further investigation if not
********************************************************************
Object ID           : xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
App ID              : xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
Display Name        : Office 365 Exchange Online
Key Credentials     :

CustomKeyIdentifier :
EndDate             : 12/9/2017 2:10:29 AM
KeyId               : xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
StartDate           : 12/9/2015 1:40:30 AM
Type                : AsymmetricX509Cert
Usage               : Verify
Value               :
```

⚠️ The script has identified a first-party (Microsoft) Service Principal with added credentials. First-party Service Principals should *not* have added credentials except in rare cases. Environments that are or were previously in a hybrid-mode may have credentials added to Exchange Online, Skype for Business, and AAD Password Protection Proxy Service Principals. This may also be an artifact of UNC2452 activity in your environment. Consult with your administrators and search the audit logs to verify the credential is legitimate. You can also use the "Service Principal Sign-Ins" tab in the Azure AD Sign-Ins blade to search for authentications to your tenant using this Service Principal.

```
 !! Identified Service Principals with high-risk API permissions and added credentials.
Verify that the added credential has a legitimate use case and consider further investigation if not
Object ID           : xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
App ID              : xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
Display Name        : TestingApp
Key Credentials     :
    CustomKeyIdentifier :
    EndDate             : 1/7/2025 12:00:00 AM
    KeyId               : xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
    StartDate           : 1/7/2021 12:00:00 AM
```

```
    Type                : Symmetric
    Usage               : Verify
    Value               :
Password Credentials :
Risky Permissions    : Domain.ReadWrite.All
```

⚠️ The script has identified a Service Principal with high-risk API permissions and added credentials. This may be expected, as some third-party or custom-built applications require added credentials in order to function. This may also be an artifact of UNC2452 activity in your environment. Consult with your administrators and search the audit logs to verify the credential is legitimate. You can also use the "Service Principal Sign-Ins" tab in the Azure AD Sign-Ins blade to search for authentications to your tenant using this Service Principal.

Applications (Invoke-MandiantAuditAzureADApplications)

This module uses Azure AD PowerShell to look for and audit Applications in Azure AD.

- **Applications with high level privileges and added credentials** - Alerts on Applications that have high-risk API permissions and added credentials. While the Applications and added permissions are likely legitimate, the added credentials may not be. Verify that the Application credentials are part of a legitimate use case. Verify that the Applications needs the listed permissions.

Example

```
 !! High-privileged Application with credentials found.
Validate that the application needs these permissions.
Validate that the credentials added to the application are associated with a legitimate use case.

ObjectID: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
AppID: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
DisplayName: Acme Test App
KeyCredentials:
PasswordCredentials:

CustomKeyIdentifier :
EndDate             : 12/22/2021 4:01:52 PM
KeyId               : xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
StartDate           : 12/22/2020 4:01:52 PM
Value               :

CustomKeyIdentifier :
EndDate             : 12/21/2021 6:32:54 PM
KeyId               : xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
StartDate           : 12/21/2020 6:33:16 PM
Value               :

Risky Permissions:
```

```
Mail.Read (Read mail in all mailboxes)
Directory.Read.All (Read all data in the organization directory)
```

⚠️ The script has identified an Application with high-risk API permissions and added credentials. This may be expected, as some third-party or custom-built applications require added credentials in order to function. This may also be an artifact of UNC2452 activity in your environment. Consult with your administrators and search the audit logs to verify the credential is legitimate.

Cloud Solution Provider Program (Invoke-MandiantGetCSPInformation)

This module checks to see if the tenant is managed by a CSP, or partner, and if delegated administration is enabled. Delegated administration allows the CSP to access a customer tenant with the same privileges as a Global Administrator. Although the CSP program enforces strong security controls on the partner's tenant, a threat actor that compromises the CSP may be able to access customer environments. Organizations should verify if their partner needs delegated admin privileges and remove it if not. If the partner must maintain delegated admin access, consider implementing Conditional Access Policies to restrict their access.

Organizations can check and manage partner relationships by navigating to the Admin Center and navigating to `Settings` -> `Partner Relationships` on the left-hand menu bar.

Mailbox Folder Permissions (Get-MandiantMailboxFolderPermissions)

This module audits all the mailboxes in the tenant for the existance of suspicious folder permissions. Specifically, this module will examine the "Top of Information Store" and "Inbox" folders in each mailbox and check the permissions assigned to the "Default" and "Anonymous" users. Any value other than "None" will result in the mailbox being flagged for analysis. In general the Default and Anonymous users should not have permissions on user inboxes as this will allow any user to read their contents. Some organizations may find shared mailboxes with this permission, but it is not recommended practice.

Application Impersonation (Get-MandiantApplicationImpersonationHolders)

This module outputs the list of users and groups that hold the ApplicationImpersonation role. Any user or member of a group in the output of this command can use impersonation to "act as" and access the mailbox of any other user in the tenant. Organizations should audit the output of this command to ensure that only expected users and groups are included, and where possible further restrict the scope.

Purview Audit (Formerly Advanced Audity) (Invoke-MandiantCheckAuditing)

This module will enumerate all licensed users in the tenant that are licensed for Purview Audit Mail Items Accessed. It will generate a CSV report documenting whether or not the feature has been enabled on an eligible mailbox. Organizations should filter on mailboxes that are eligible for Mail Items Accessed but have the feature disabled and verify that this is intentional.

Unified Audit Log (Get-MandiantUnc2452AuditLogs)

This module is a helper script to search the Unified Audit Log. Searching the Unified Audit Log has many technical caveats that can be easy to overlook. This module can help simplify the search process by implementing best practices for navigating these caveats and handling some common errors.

By default, the module will search for log entries that can record UNC2452 techniques. The log records may also capture legitimate administrator activity, and will need to be verified.

- **Update Application** - Records actions taken to update App Registrations.
- **Set Domain Auth** - Records when authentication settings for a domain are changed, including the creation of federation realm objects. These events should occur rarely in an environment and may indicate a threat actor configuring an AAD backdoor.
- **Set Federation Settings** - Records when the federation realm object for a domain is modified. These events should occur rarely in an environment and may indicate a threat actor preparing to execute a Golden SAML attack.
- **Update Application Certificates and Secrets** - Records when a secret or certificate is added to an App Registration.
- **PowerShell Mailbox Logins** - Records Mailbox Login operations where the client application was PowerShell.
- **Update Service Principal** - Records when updates are made to an existing Service Principal.
- **Add Service Principal Credentials** - Records when a secret or certificate is added to a Service Principal.
- **Add App Role Assignment** - Records when an App Role (Application Permission) is added.
- **App Role Assignment for User** - Records when an App Role is assigned to a user.
- **PowerShell Authentication** - Records when a user authenticates to Azure AD using a PowerShell client.
- **New Management Role Assignments** - Records when new management role assignments are created. This can be useful to identify new ApplicationImpersonation grants.

Usage

Required Modules

The PowerShell module requires the installation of three Microsoft 365 PowerShell modules.

- AzureAD
- MSOnline
- ExchangeOnlineManagement
- Microsoft.Graph

To install the modules:

1. Open a PowerShell window as a local administrator (right-click then select Run As Administrator)
2. Run the command `Install-Module <MODULE NAME HERE>` and follow the prompts

Required User Permissions

The PowerShell module must be run with a Microsoft 365 account assigned specific privileges.

- `Global Administrator` or `Global Reader` role in the Azure AD portal
- `View-Only Audit Logs` in the Exchange Control Panel
- `User.Read.All` and `Directory.Read.All` scopes. `Global Reader` role holders should have the ability to use these scopes automatically.

To grant an account `View-Only Audit Logs` in the Exchange Control Panel:

1. Navigate to https://outlook.office365.com/ecp and login as a global admin or exchange admin (not the exact URL may differ if you are in an alternate cloud)
2. Click `admin roles` in the dashboard, or expand the `roles` tab on the left and click `admin roles` if you are in the new UI
3. Create a new admin role by clicking the `+` sign or clicking `add new role group`
4. Give your role a name and default write-scope
5. Add the `View-Only Audit Logs` permission to the role
6. Add the user to the role

Note it can take up to an hour for this role to apply

Running the tool

1. Download this tool as a ZIP and unzip it, or clone the repository to your system
2. Open a PowerShell window
3. Change directories to the location of this module `cd C:\path\to\the\module`
4. Import this module `Import-Module .\MandiantAzureADInvestigator.psd1` you should receive this output

```
      Mandiant Azure AD Investigator
    Focusing on UNC2452 Investigations

PS C:\Users\admin\Desktop\mandiant>
```

5. Connect to Azure AD by running `Connect-MandiantAzureEnvironment -UserPrincipalName <your username here>`. You should receive a login prompt and output to the PowerShell window indicating the connections have been established. **Note**: If you run into issues you may need to change your execution policy by running `Set-ExecutionPolicy -ExecutionPolicy RemoteSigned`. This may require administrator privileges.

```
 ---------------------------------------------------------------------------
The module allows access to all existing remote PowerShell (V1) cmdlets in addition to the 9 new, faster, and mo

|-------------------------------------------------------------------------|
|    Old Cmdlets                    |    New/Reliable/Faster Cmdlets      |
|-------------------------------------------------------------------------|
|    Get-CASMailbox                 |    Get-EXOCASMailbox                |
|    Get-Mailbox                    |    Get-EXOMailbox                   |
```

```
|    Get-MailboxFolderPermission    |    Get-EXOMailboxFolderPermission    |
|    Get-MailboxFolderStatistics    |    Get-EXOMailboxFolderStatistics    |
|    Get-MailboxPermission          |    Get-EXOMailboxPermission          |
|    Get-MailboxStatistics          |    Get-EXOMailboxStatistics          |
|    Get-MobileDeviceStatistics     |    Get-EXOMobileDeviceStatistics     |
|    Get-Recipient                  |    Get-EXORecipient                  |
|    Get-RecipientPermission        |    Get-EXORecipientPermission        |
|-----------------------------------------------------------------------------|

To get additional information, run: Get-Help Connect-ExchangeOnline or check https://aka.ms/exops-docs

Send your product improvement suggestions and feedback to exocmdletpreview@service.microsoft.com. For issues re
------------------------------------------------------------------------------

Account                                           Environment TenantId                            TenantDoma
-------                                           ----------- --------                            ----------
doug@test.onmicrosoft.com AzureCloud  xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx test.onm...
```

5. Run all checks `Invoke-MandiantAllChecks -OutputPath <path\to\output\files>`. You can also run individual checks using the specific cmdlet.
6. Review the output on the screen and the written CSV files.

Further Reading

For additional information from Mandiant regarding UNC2452, please see:

- Highly Evasive Attacker Leverages SolarWinds Supply chain to Compromise Multiple Global Victims with SUNBURST Backdoor
- Remediation and Hardening Strategies for Microsoft 365 to Defend Against UNC2452

The response to UNC2452 has been a significant effort across the security industry and these blogs heavily cite additional contributions that will be of value to users of this tool. We recommend reading the linked material from these posts to best understand activity in your environment. As always, the Mandiant team is available to answer follow-up questions or further assist on an investigation by contacting us here.
