# Audit des permissions en environnement Active Directory

Géraud de Drouas et Pierre Capillon
geraud.de-drouas(@)ssi.gouv.fr
pierre.capillon(@)ssi.gouv.fr

Agence Nationale de la Sécurité des Systèmes d'Information

**Résumé** Lors de l'audit d'un environnement Active Directory, les permissions mises en œuvre par le mécanisme de contrôle d'accès discrétionnaire sont le plus souvent examinées de manière sommaire, du fait de leur nombre et des limites inhérentes aux outils intégrés au système. Les privilèges illégitimes ou inadaptés acquis via les permissions de l'annuaire doivent pouvoir être détectés afin de prévenir les risques d'abus ou d'escalade mais aussi de retour ou de persistance d'un attaquant au sein d'un système d'information compromis après remise en état.

La principale contribution de cet article est de proposer une approche pratique d'audit de l'ensemble des permissions d'un environnement Active Directory. Nous expliquerons le modèle de contrôle d'accès et détaillerons une méthode de récupération des descripteurs de sécurité depuis les fichiers de base de données de l'annuaire. Enfin l'outillage développé pour visualiser et analyser ces informations sera présenté.

## 1 Introduction

La protection des ressources informationnelles d'une organisation repose sur un modèle de contrôle d'accès. Celui-ci doit assurer l'authentification, l'autorisation et la journalisation des transactions effectuées par les utilisateurs vis-à-vis des différentes ressources. L'audit d'un système de contrôle d'accès se contentant d'un simple examen de la robustesse de l'authentification serait donc incomplet.

La mise en œuvre technique d'une solution de contrôle d'accès dans des environnements dominés par les systèmes Windows passe généralement par le déploiement d'une architecture Active Directory. Elle fournit l'authentification en environnement Windows, mais sert également de référence pour la mise en œuvre du modèle de contrôle d'accès discrétionnaire, qui est au cœur de la sécurité du monde Windows. Chaque ressource porte son propre descripteur de sécurité, dans lequel se trouvent les utilisateurs et groupes qui ont légitimement besoin d'y accéder et ceux qui en sont explicitement empêchés.

Ce modèle de permissions, dans le cadre de la compromission d'un Active Directory, peut être utilisé pour établir des chemins persistants d'élévation et de maintien de privilèges d'accès aux ressources, grâce à la seule logique de l'annuaire, de manière décorrélée d'une précédente exploitation de vulnérabilité système. Différents types de privilèges techniques permettent indirectement l'accès aux ressources informationnelles de l'organisation, par le jeu des héritages de permissions et de la manipulation des titulaires de permissions eux-mêmes. Il s'agit donc d'identifier puis d'auditer les objets Active Directory dont le contrôle permet indirectement l'accès aux ressources à protéger.

Bien qu'il soit possible de consulter les permissions manuellement au moyen de l'interface graphique des systèmes Windows, cette routine est longue et répétitive ; un auditeur souhaitera l'automatiser en utilisant directement la base de données de l'annuaire Active Directory. Par ailleurs, seul le point de vue discrétionnaire, propre à chaque ressource, est disponible via l'interface graphique et les outils du système, alors qu'il est souhaitable de rechercher aussi tous les titulaires d'un type de permission précis ou bien toutes les permissions accordées à un titulaire particulier.

Une fois les données nécessaires extraites d'Active Directory, l'approche proposée passe par leur réinjection dans une base de données dont la structure permet des requêtes pertinentes pour l'auditeur. Une solution de visualisation et d'aide à l'analyse a été réalisée pour automatiser ces requêtes et filtrer leurs résultats.

Cet article ne se borne pas à présenter un outil, mais détaille les différentes réflexions et axes d'améliorations envisagés pour une méthodologie d'audit efficace sur des bases Active Directory à gros volume d'information. Après une explication des mécanismes théoriques et de l'implantation technique des mécanismes de contrôle d'accès, seront abordés les points à considérer dans le cadre d'un audit classique ou de l'analyse d'une compromission. Enfin seront détaillés les besoins en outillage pour réaliser une approche d'audit exhaustive, les défis techniques rencontrés et les solutions trouvées ou envisagées.

## 2 Considérations méthodologiques pour l'audit d'un Active Directory

Pour réaliser l'audit des permissions présentes dans la base de l'Active Directory, la méthodologie employée conjugue plusieurs approches complémentaires :

- l'analyse des déviations par rapport aux permissions standards ou préconisées par Microsoft ;
- l'analyse de certains objets spéciaux de l'Active Directory dont le fonctionnement est souvent peu documenté ;
- l'analyse globale des objets pour déterminer les titulaires de certains droits plus ou moins dangereux ;
- l'analyse centrée sur les types de permissions accordées, ce que l'interface graphique du système ne permet pas ;
- l'analyse centrée sur les titulaires de droits, également non disponible par l'interface graphique. Cette démarche révèle l'ensemble des permissions possédées par un titulaire et l'ensemble des objets concernés. Elle permet de mettre en évidence les comptes possédant, par exemple, des droits sur bien plus d'objets que nécessaire, voire sur des objets critiques permettant la prise de contrôle du domaine.

Un outillage est alors nécessaire pour mettre en œuvre ces différentes approches et synthétiser les informations utiles pour l'auditeur. Il s'agira, entre autres, de résoudre dynamiquement les appartenances à des groupes d'utilisateurs pour déterminer les permissions réellement applicables à tel ou tel compte, en plus de présenter les informations d'une manière facilitant l'audit par objet, par titulaire ou par type de permission. Enfin, l'outil pourra intégrer des fonctionnalités de rejeu de critères de recherche d'un audit à l'autre, alimenter une base de connaissance interne, ou intégrer les droits par défaut des installations « fraîches » d'un domaine Active Directory, en fonction de sa version ou des différents scénarios de migration. Cette démarche permet à l'auditeur d'être bien plus efficace et exhaustif que lors d'un audit basé sur les permissions rencontrées au cas par cas ou découvertes via l'interface graphique de manière unitaire sur des objets particuliers de l'annuaire.

## 3 Mécanismes d'un Active Directory

### 3.1 Contrôle d'accès

**Généralités**

Un mécanisme de contrôle d'accès assure les fonctions d'authentification, d'autorisation et de journalisation lors de l'accès d'une entité à une ressource.

**Modèle DACL**

Le contrôle d'accès discrétionnaire est essentiellement basé sur la notion

de propriété. Chaque ressource est protégée individuellement, selon des autorisations placées par le propriétaire de l'objet vis-à-vis d'autres entités du système. Le propriétaire initial est le plus souvent le créateur de l'objet, mais la propriété peut être transférée. Ce transfert peut être manuel, ou bien automatique comme dans le cas d'ajouts d'éléments dans l'annuaire par un administrateur, dont le propriétaire devient le groupe `Admins de domaine` et non plus le créateur.

Toute ressource protégeable par le contrôle d'accès discrétionnaire Microsoft est appelée « `Securable Object` ». Quels que soient son type et son mode de stockage, elle dispose systématiquement d'un descripteur de sécurité qui lui est propre. Les objets du système de fichiers NTFS, les clés de la base de registre, les objets de l'Active Directory sont autant d'exemples de « `Securable Objects` ». Un descripteur de sécurité contient notamment la liste de contrôle d'accès (ACL, « *Access Control List* »), composée d'entrées décrivant chaque permission accordée ou refusée : les « *Access Control Entries* » (ACE).

Toute entité susceptible d'être titulaire d'une permission (utilisateur, machine, etc.) est appelée « `Security Principal` ». Dans le modèle Microsoft, elle est désignée par un SID [1] (« `Security IDentifier` »).

**Mécanisme d'autorisation**

Le plus souvent, les ressources sont créées dans un conteneur, au sein d'une hiérarchie correspondant à leur contexte et à leur type, par exemple une arborescence de répertoires dans un système de fichiers ou la structure de la base de registre. La hiérarchie se traduit en un concept d'héritage du contrôle d'accès. Les permissions appliquées aux éléments parents vont généralement être incluses dans le descripteur de sécurité des éléments enfants lors de leur création, ou lors de la modification du descripteur du parent. L'héritage peut être bloqué sur l'élément enfant par un paramétrage de ce dernier ou si l'ACE est marquée comme non héritable.

Le propriétaire d'un objet dans le modèle Microsoft est appelé « `primaryOwner` » et possède par défaut les privilèges lui permettant de lire et de modifier le descripteur de sécurité. Par conséquent, le propriétaire peut tout à fait ne pas disposer de permissions apparentes sur une ressource et se les accorder au besoin.

La vérification de l'autorisation [8] s'effectue de manière séquentielle en suivant la liste de contrôle d'accès portée par une ressource via son descripteur de sécurité, selon les droits demandés [4]. Un processus demandeur présente généralement plusieurs SID dans son jeton d'accès, comme

1. Des listes de SID connus sont disponibles en ligne [13,12].

celui de l'utilisateur sous l'identité duquel le processus tourne et ceux des groupes auxquels cet utilisateur appartient :

- seules les ACE dont le SID titulaire est présent et actif dans le jeton d'accès du demandeur sont conservées ;
- chaque ACE restante est considérée à son tour et valide individuellement les bits de l'`Access Mask` demandé, à partir de son propre `Access Mask` (voir la section 4.3) ;
- le mécanisme s'arrête dès que tous les bits de l'`Access Mask` demandé sont validés, ce qui autorise l'accès. Si le mécanisme prend fin parce que toutes les ACE ont été inspectées, l'accès est refusé ;
- si une ACE d'interdiction valide un bit demandé, l'accès est refusé immédiatement.

## 3.2 Points d'intérêt lors de l'audit

**Objets particuliers**

Certains types d'objets nécessitent une attention particulière dans l'audit des permissions à cause de leurs spécificités dans le modèle de contrôle d'accès :

- l'objet de protection : `adminSDholder`. Il sert de séquestre de descripteur de sécurité pour les groupes dits « protégés ». Le fonctionnement très particulier de cet objet est détaillé plus bas, dans la sous-section qui lui est dédiée ;
- les comptes « *builtin* », installés à la création de l'Active Directory : Administrateur, mais aussi `krbtgt`, le compte dont le condensat `NTHash` est le secret du contrôleur de domaine dans sa fonction d'`Authentication Server` Kerberos. Il sert à signer les TGT (`Ticket Granting Tickets`), éléments d'authentification du protocole Kerberos ;
- les groupes « *builtin* ». Certains sont naturellement puissants et permettent la prise de contrôle du domaine, tels que les administrateurs de domaine ou les opérateurs de comptes. D'autres semblent moins dangereux, comme les opérateurs de sauvegarde ou les opérateurs d'impression, mais tous disposent du privilège d'ouverture de session sur les contrôleurs de domaine, donc de la possibilité d'y exécuter du code ;
- les unités organisationnelles (OU), pour lesquelles sont attribuées des délégations d'administration. Elles constituent un niveau supplémentaire de prise de contrôle possible, moins visible que les groupes d'administration du domaine. Elles permettent la prise de contrôle

de leurs objets enfants, utilisateurs en particuliers, donc indirectement des ressources auxquelles ces objets ont accès ;
- les entités disposant de permissions de délégation d'authentification. Par défaut, seuls les contrôleurs de domaine peuvent s'authentifier à la place d'une autre entité sur le domaine, néanmoins cette fonctionnalité est proposée par le protocole Kerberos pour les services ayant besoin de ressources externes appartenant à leurs clients. Correctement configurée, cette possibilité est restreinte aux ressources désignées par leur SPN (`Service Principal Name`), sous la forme "Service/FQDN", dans les attributs du titulaire. Si le SPN choisi est particulièrement critique ou que la permission n'est pas limitée, les attaques par réflexion deviennent possibles.

Le recensement de ces types d'objets peut se faire par leur attribut `Object-Category`, récupérable dans la base. Il contient un entier, variable selon les domaines, renvoyant à la propriété `Default-Object-Category`, d'où l'on peut déduire le type d'objet auquel on a affaire. De cette manière, on garantit l'exhaustivité de l'audit des types d'objets ciblés.

**Le cas particulier de l'objet `adminSDholder`**

Un processus particulier, `SDProp` (*Security Descriptor Propagator*), écrase périodiquement le descripteur de sécurité de tous les membres des groupes dits « protégés » par celui de l'`adminSDholder`. Une modification de ce descripteur peut être très discrète si elle sert à reprendre périodiquement et brièvement le contrôle de comptes d'administration via `SDProp`, puis à restaurer les ACE légitimes immédiatement après.

Cet objet particulier est peu documenté par Microsoft [9] et son existence est souvent méconnue, voire ignorée des administrateurs de domaines. Il est visible sous le *distinguished name* `CN=AdminSDHolder,CN=System,DC=<doma` Il en existe donc une instance pour chaque domaine et sous-domaine d'une forêt, qui ne s'applique qu'à la branche à laquelle il appartient.

Le mécanisme est justifié par le besoin de protection des administrateurs vis-à-vis d'eux-mêmes, afin d'éviter qu'ils ne se retirent par maladresse des droits essentiels à la bonne gestion du système d'information.

Le processus *SDProp* réapplique périodiquement les ACE de `adminSDholder` (par défaut toutes les heures) aux utilisateurs et groupes suivants [2] entre autres [11] :
- groupes « *builtin* » et membres de ces groupes de l'AD :
  - Administrateurs,

2. La protection de ces groupes peut être inhibée unitairement par l'attribut `dsHeuristics` [10].

  - Administrateurs du schéma,
  - Administrateurs du domaine,
  - Administrateurs de l'entreprise,
  - Opérateurs de comptes,
  - Opérateurs de sauvegarde,
  - Opérateurs d'impression ;
- certains comptes particuliers :
  - administrateur,
  - krbtgt,
  - comptes des contrôleurs de domaines (DC et RO-DC [3]).

Il est donc particulièrement important d'auditer d'une part les membres de ces groupes, mais surtout les ACE portées par l'objet `adminSDholder` lui-même, pour s'assurer qu'aucune ACE n'ait été ajoutée par malveillance pour octroyer des droits illégitimes à un attaquant, sur les groupes et utilisateurs listés précédemment.

C'est un moyen de persistance puissant puisque les autorisations sont réappliquées automatiquement et plus ou moins furtivement par le système. On pourrait imaginer ainsi l'ajout d'une autorisation d'écriture de propriété, donnée à un utilisateur particulier, lui permettant de changer à volonté l'attribut `Member` des groupes d'administration « protégés » pour y rajouter le compte de son choix. L'état des ACL des groupes d'administration paraîtra, entre chaque rafraîchissement par `SDProp`, parfaitement normal.

L'interface graphique (figure 1) ne prévient pas l'administrateur lors de l'édition d'un compte impacté par ce mécanisme. Ironiquement, l'équipe *Directory Services* explique sur son blog Technet [4] qu'il faut en être conscient : « *–Is there a way to warn administrators that a user being manipulated is covered under AdminSDHolder and SDProp ? –Nope, you just gotta know.* ».

### 3.3 Droits étendus, cas d'Exchange

L'Active Directory a été conçu comme un annuaire généraliste, même s'il est essentiellement utilisé pour l'administration des domaines Windows. Le schéma originel peut être étendu par divers attributs et propriétés, pour les besoins d'applications supplémentaires. Ainsi, Microsoft Exchange utilise largement sur cette possibilité. Le fonctionnement appli-

3. *Read-only domain controller.*
4. Voir : http://blogs.technet.com/b/askds/archive/2009/05/07/five-common-questions-about-adminsdholder-and-sdprop.aspx.

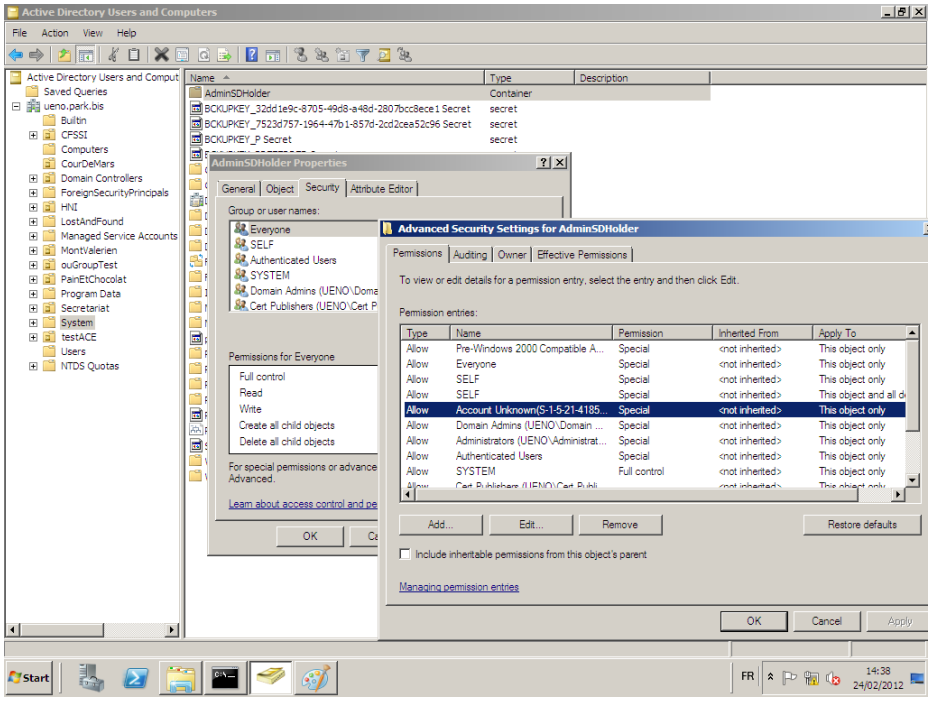



**Figure 1.** Les ACE portées par l'objet `adminSDholder` vues dans l'interface graphique standard

catif peut requérir des permissions particulières, reflétées dans le descripteur de sécurité par l'utilisation des droits étendus. Dans une telle ACE, le bit 0x100 de l'`AccessMask` est positionné et le champ `Object-Type` contient un GUID désignant un droit propre à l'application. Par exemple, `Receive-As` est un droit étendu Exchange permettant l'accès en lecture à une boîte aux lettres, `Send-As` est le droit étendu permettant d'envoyer des courriers sous l'identité du compte auquel est liée la boîte.

Ces permissions sont largement utilisées dans les cas de boîtes aux lettres déléguées à des secrétaires ou de boîtes partagées. La messagerie étant une ressource informationnelle particulièrement riche, ces types de permissions sont appréciés des attaquants. L'utilisation d'outils automatisés de migration ou de traitement des permissions par les administrateurs ayant tendance à multiplier les ACE dans l'architecture Exchange, donc dans les boîtes aux lettres, il est parfois difficile d'identifier la légitimité de droits d'accès externes sur une boîte donnée. Notre approche permet de révéler les titulaires de droits sur des boîtes aux lettres multiples qui ne leur appartiennent pas, donc suspects.

Les boîtes aux lettres disposent de permissions qui leur sont propres et de permissions héritées, de par l'arborescence des objets Exchange au sein d'Active Directory, qui fonctionne comme suit pour la version 2010 :

```
‘- Partition de configuration
   +- Services
      +- Microsoft Exchange
         +- <Nom de l'organisation>
            +- Groupes administratifs
               +- <Nom du groupe administratif>
                  +- Databases
                  |  +- <Nom de la base de donnees>
                  |     +- <Nom du serveur MBD>
                  |        +- Boites aux lettres
                  +- Folder Hierarchies
                     +- Public Folders
                        +- <Nom du repertoire public>
```

Chacun de ces objets doit donc être audité.

## 4 Travailler avec la base de données

### 4.1 La technologie ESE

Plusieurs articles ont été publiés cette année autour d'ESE, `Extensible Storage Engine`, la technologie de base de données utilisée par Active Directory, en particulier sur le sujet de la récupération des condensats de mots de passe [1]. ESE est une technologie de base de données non relationnelle, intégrée aux systèmes Windows depuis Windows 2000. Anciennement appelée Jet Blue, à ne pas confondre avec Jet Red, technologie utilisée par Microsoft Access. Elle utilise des fichiers binaires à plat dont la taille peut atteindre 16 To, contenant potentiellement plusieurs tables.

ESE ne possède pas de fournisseur de requêtes ; charge à chaque application de mettre en place la solution de requête dont elle a besoin. La technologie ne fournit que le stockage brut des données. La lecture et la modification s'effectuent uniquement avec des index associés aux différentes colonnes ou par simple accès séquentiel. Les performances en insertion et extraction brutes de données sont donc beaucoup plus rapides qu'avec d'autres bases de données relationnelles, mais le service fourni est comparativement limité.

L'API fournie par Microsoft est d'assez bas niveau. Elle est exportée par la bibliothèque `esent.dll` et convenablement documentée dans la MSDN, bien que peu d'exemples d'utilisation existent sur Internet. Quelques projets ont vu le jour afin d'abstraire une partie des transactions nécessaires à l'exploitation des bases et de proposer un fournisseur de requêtes, tel que `libesedb` [2].

L'API est néanmoins tout à fait utilisable en l'état, à condition de savoir se contenter des extractions de données séquentielles ou indexées. L'interface graphique reflète ces choix d'architecture par les différentes lacunes évoquées en introduction. Cette limitation n'étant pas acceptable pour une approche d'audit qui nécessite de nombreux croisements, il est nécessaire d'utiliser une autre base de données. L'Active Directory, en pratique, effectue ces opérations nécessaires à son fonctionnement au cours de son exécution.

### 4.2 Récupération de la base en pratique

Obtenir le contenu de l'annuaire Active Directory peut se faire de différentes manières, chacune ayant ses avantages et ses inconvénients. La plus simple est la connexion en protocole LDAP. Le service est accessible par tout client respectant la RFC 4510. Après un `bind` avec les identifiants d'un compte possédant les privilèges nécessaires, il est possible de lister les objets par partition puis par unité d'organisation, pour ensuite en extraire tous les attributs. Certains attributs ne peuvent cependant pas être obtenus de cette manière, en particulier les « attributs protégés » de l'annuaire, qui comprennent condensats de mots de passe chiffrés, clés de recouvrement BitLocker et autres secrets.

La connexion aux LPC de réplication inter-contrôleurs de domaine est une seconde solution, qui permet d'obtenir l'ensemble des attributs, mais nécessite d'interpréter le format de réplication.

Ces deux méthodes nécessitent l'accès à un contrôleur de domaine en ligne, ce qui peut poser des problèmes de charge, aussi bien pour les performances générales du contrôleur que pour la vitesse de récupération elle-même. Cela peut devenir contraignant dans les cas de quantités massives de données, ce qui est le cas des descripteurs de sécurité, point central de la démarche d'audit présentée ici.

Une troisième solution est l'extraction des données hors ligne à partir du fichier de la base de données Active Directory, `ntds.dit`. En raison de son ouverture exclusive par le système, ce fichier ne peut être copié que depuis un contrôleur de domaine éteint, ou à partir d'une reconstruction de raid ou bien d'un cliché de sauvegarde réalisé avec le `Volume Shadowcopy Service`. À partir des systèmes Windows 2000, l'API `ntbackup` assure ce rôle. C'est cette solution que nous utilisons pour l'audit des permissions. Elle demande un travail supplémentaire pour retrouver certains attributs, qui sont reconstruits par l'agent NTDSA à l'exécution.

Dans les cas de copie du fichier en ligne, c'est-à-dire pendant que la base de données est en cours d'utilisation, il faudra y joindre les

fichiers de journalisation et de point de contrôle pour en réaliser la fermeture complète et propre, avec les utilitaires fournis par l'éditeur, tel que « `esentutl.exe` ». Ces fichiers sont tous, par défaut, dans `C:\Windows\NTDS`. Si le répertoire a été modifié lors de la promotion du contrôleur de domaine, on le retrouve dans la clé de registre `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\NTDS\Parameters`. Le listing 1.1 montre le déroulement d'une copie complète des fichiers selon cette méthode.

```
C:\Windows\system32>vssadmin create shadow /for=C:
Successfully created shadow copy for 'C:\'
    Shadow Copy ID: {e043819e-3609-4527-ba05-58e3336910ff}
    Shadow Copy Volume Name: \\?\GLOBALROOT\Device\
        HarddiskVolumeShadowCopy5

C:\Windows\system32>copy \\?\GLOBALROOT\Device\
    HarddiskVolumeShadowCopy5\Windows
\NTDS C:\adpop\ntds2
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy5\Windows\NTDS\edb.chk
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy5\Windows\NTDS\edb.log
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy5\Windows\NTDS\
    edb00017.log
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy5\Windows\NTDS\
    edb00018.log
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy5\Windows\NTDS\
    edbres00001.jrs
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy5\Windows\NTDS\
    edbres00002.jrs
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy5\Windows\NTDS\ntds.
    dit
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy5\Windows\NTDS\temp.
    edb
        8 file(s) copied.

C:\Windows\system32>copy \\?\GLOBALROOT\Device\
    HarddiskVolumeShadowCopy5\Windows
\System32\config\system C:\adpop\ntds2
        1 file(s) copied.

C:\Windows\system32>vssadmin delete shadows /for=C:
Do you really want to delete 5 shadow copies (Y/N): [N]? Y
Successfully deleted 5 shadow copies.
```

**Listing 1.1.** Création d'un cliché et récupération des bases ESE

La fermeture du fichier se fait en spécifiant la taille de page de la base, 8ko dans le cas d'Active Directory : `esentutl.exe /r edb /d /i /8`. Le listing 1.2 présente l'opération.

```
C:\adpop\ntds2>esentutl.exe /r edb /d /i /8
Initiating RECOVERY mode...
    Logfile base name: edb
```

```
            Log files: <current directory>
         System files: <current directory>
   Database Directory: <current directory>
Performing soft recovery...
                      Restore Status (% complete)
          0    10   20   30   40   50   60   70   80   90  100
          |----|----|----|----|----|----|----|----|----|----|
          ..................................................
Operation completed successfully in 0.984 seconds.
```

**Listing 1.2.** Fermeture du fichier ESE par exécution des transactions en attente

L'outil développé spécifiquement pour l'extraction des bases ESE peut alors être lancé pour obtenir les données des tables concernant les attributs, ACE et descripteurs de sécurité (voir listing 1.3).

```
C:\adpop\ntds2>esent_dump.exe
Usage: esent_dump [option] <ntdsfile>
Options (mutually exclusive):
        ad: dump the whole AD datatable
        sid: dump the Common-Name/OU-Name to SID, Object-Category,
            NT SD and Mailbox SD index resolution
        ace: dump the AD ACE from sd_table
        att : dump column naming columns
        cat : dump object categories number to description

C:\adpop\ntds2>esent_dump.exe sid ntds.dit
[*]Table MSysObjects ouverte
[*]datatable tableID found : 8
[*]Table datatable ouverte
Dumping datatable column names...
Dumping content...
cleaning...

C:\adpop\ntds2>esent_dump.exe att ntds.dit
[*]Table MSysObjects ouverte
[*]datatable tableID found : 8
[*]Table datatable ouverte
Dumping datatable column names...
Dumping content...
cleaning...
```

**Listing 1.3.** Extraction des données de la base Active Directory

### 4.3 Structure de la base

Dans le fichier `ntds.dit`, trois tables vont plus particulièrement nous servir :

– `MSysObjects` : la table des métadonnées, essentielle pour lier les identifiants de colonnes aux types de données contenues ;

- `datatable` : la table des objets Active Directory. Tout ce qui constitue les partitions de l'annuaire visible en LDAP se retrouve là : objets du domaine, configuration, schéma, DNS, etc. ;
- `sd_table` : la table des descripteurs de sécurité des objets sécurisables contenus dans l'annuaire Active Directory.

Chaque objet de l'Active Directory occupe une ligne dans la table `datatable`. Utilisateurs et ordinateurs, mais aussi éléments du schéma, objets de la configuration sont stockés de cette manière. Chaque colonne est un attribut ou une propriété définie dans le schéma. De nombreux attributs apparaissent vides lors de l'extraction du contenu de la table, ceci dépendant des propriétés s'appliquant à chaque type d'objet. Par ailleurs, de nombreux attributs sont reconstruits par l'agent NTDSA lors de son exécution. En particulier, c'est le cas de `MemberOf`, reconstruit à partir des attributs `Member`, ou bien du `Distinguished-Name`, reconstruit à partir d'une colonne contenant des numéros d'ancêtres.

**Récupération des attributs**

L'ouverture d'une table peut se faire directement avec l'API ESENT de la manière suivante, assez peu documentée sur Internet :

```
JetSetSystemParameter(0, JET_sesidNil, JET_paramDatabasePageSize,
    8192, NULL);
JetCreateInstance(&instance, "monInstance");
JetInit(&instance);
JetBeginSession(instance, &sesid, 0, 0);
JetAttachDatabase(sesid, baseName, JET_bitDbReadOnly);
JetOpenDatabase(sesid, baseName, 0, &dbid, 0);
JetOpenTable(sesid, dbid, tableName, 0, 0, JET_bitTableReadOnly, &
    tableid);
```

Le parcours des enregistrements d'une table est généralement très rapide, l'opération fonctionnant simplement avec un curseur. On utilise pour cela :

```
JetMove(sesid, tableid, JET_MoveFirst, 0);
```

et

```
JetMove(sesid, tableid, JET_MoveNext, 0)
```

Les attributs sont stockés dans des colonnes qui doivent être récupérées au moyen de l'API ESENT. Pour cela, il faut lire les méta-données dans la table `MSysObject`. On obtient l'ID de la table souhaitée (`datatable` ou `sd_table`) en simple correspondance avec son nom. Puis les méta-données de chacune des colonnes de la table souhaitée doivent être lues de

la même manière en appelant `JetGetColumnInfo` : noms, types, identifiant de colonne, identifiant de table.

```
JetGetColumnInfo(sesid, dbid, tableName, "Id", columndefid, sizeof(
    JET_COLUMNDEF), JET_ColInfo);
JetGetColumnInfo(sesid, dbid, tableName, "Type", columndeftype,
    sizeof(JET_COLUMNDEF), JET_ColInfo);
JetGetColumnInfo(sesid, dbid, tableName, "ColtypOrPgnoFDP",
    columndeftypecol, sizeof(JET_COLUMNDEF), JET_ColInfo);
JetGetColumnInfo(sesid, dbid, tableName, "Name", columndefname,
    sizeof(JET_COLUMNDEF), JET_ColInfo);
JetGetColumnInfo(sesid, dbid, tableName, "ObjIdTable",
    columndefobjid, sizeof(JET_COLUMNDEF), JET_ColInfo);
```

Une fois ces informations obtenues, il devient possible d'extraire les données elles-mêmes dans les tables qui nous intéressent, en faisant appel à `JetRetrieveColumn`. Les informations de typage récupérées dans les méta-données seront alors utilisées pour déterminer un format de stockage intermédiaire.

L'étape suivante consiste à faire le lien entre les noms de colonnes utilisés par Active Directory et les `LDAP-Display-Name` qui représentent les attributs dans un format compréhensible par un être humain. Les noms de colonnes de la table « `datatable` » sont formatés la plupart du temps avec le préfixe « `ATT` », suivi d'une lettre minuscule indiquant généralement le type de donnée stockée dans la colonne, puis d'un nombre.

Une recherche de texte dans la table à plat, sur les différents attributs par défaut d'Active Directory – documentés dans la MSDN –, nous permet de repérer la colonne contenant très probablement les noms d'attributs : « `ATTm131532` ».

La résolution de l'indirection semble alors logique : chercher quelle colonne contient justement « `ATTm131532` » pour la ligne correspondant dans la table à l'objet de schéma « `LDAP-Display-Name` ». Il n'est pas possible de trouver cette valeur, on trouve néanmoins « `131532` » dans la colonne « `ATTc131102` ».

La figure 2 résume le processus :

1. repérage manuel de la colonne contenant tous les noms d'attributs ;
2. recherche dans cette même colonne du terme « `LDAP-Display-Name` » ;
3. sur la ligne correspondante, recherche du nom (partiel) de la colonne repérée à l'étape 1 ;
4. la nouvelle colonne où cette correspondance est faite est celle qui contient tous les noms partiels de colonnes ;
5. chaque attribut qui nous intéresse est ensuite repéré dans la colonne de l'étape 2 et son nom partiel de colonne est récupéré dans celle de

l'étape 4. Une recherche dans les noms de colonne donne alors le nom complet de colonne sans ambiguïté.

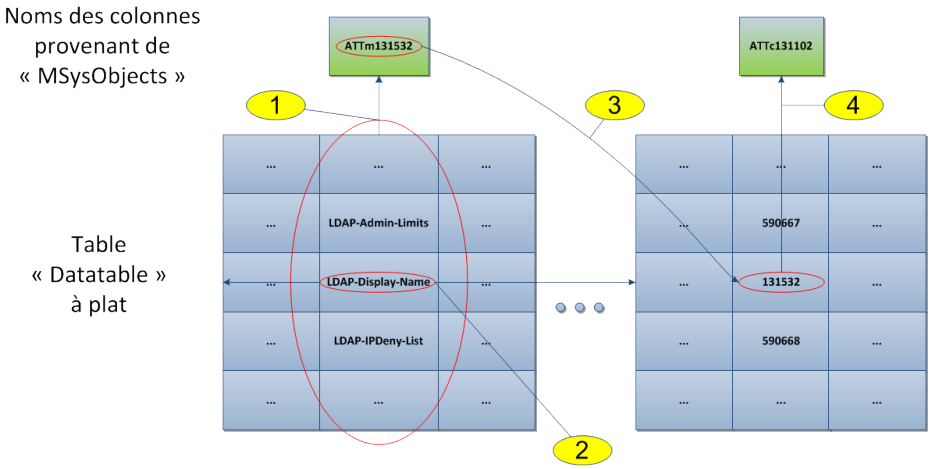



**Figure 2.** Récupération des attributs dans la table ESE

En établissant cette correspondance pour d'autres attributs LDAP du schéma, on retrouve les colonnes qui vont nous intéresser :

- « `ATTm3` » Common-Name ;
- « `ATTm11` » Organizational-Unit-Name ;
- « `ATTm1376281` » Domain-Component ;
- « `ATTm131532` » LDAP-Display-Name ;
- « `ATTm590480` » User-Principal-Name ;
- « `ATTc131102` » Attribute-ID : colonne des numéros de colonnes ;
- « `ATTj591540` » msDS-IntId : idem, mais pour des extensions de schéma spécifiques (Exchange en particulier) ;
- « `ATTm590045` » Username ;
- « `ATTr589970` » Object-Sid ;
- « `ATTb590606` » Object-Category ;
- « `ATTb590607` » Default-Object-Category ;
- « `ATTp131353` » NT-Security-Descriptor.

En pratique, on constate que l'attribut `NT-Security-Descriptor` ne contient pas de structure de descripteur de sécurité, ni binaire, ni au format SDDL comme on pourrait s'y attendre. Il contient simplement un entier, utilisable à partir du système de stockage des descripteurs de sécurité dans la base Active Directory.

Pour obtenir le descripteur de sécurité associé à un objet, il faut lire la table « `sd_table` ». Celle-ci contient en particulier deux colonnes : « `sd_id` » et « `sd_value` ». Cette dernière colonne stocke des objets binaires qui correspondent bien à la structure `SECURITY_DESCRIPTOR` définie dans le fichier d'en-tête du Software Developement Kit de Windows, `WinNT.h`. La colonne « `sd_id` » est l'index unique du descripteur de sécurité. On y trouve également une colonne « `sd_hash` ».

Ce format de stockage est dit « `single instanced` ». Il permet d'éviter le stockage redondant de descripteurs de sécurité identiques. La grande majorité des objets Active Directory possède un descripteur de sécurité par défaut correspondant à sa catégorie. Lorsqu'un nouvel objet est créé et que son descripteur est déterminé, ou lorsqu'un descripteur est modifié, il est condensé et le résultat est recherché dans la colonne `sd_hash`. Dans le cas d'un descripteur identique pré existant, son index est placé dans l'attribut `NT-Security-Descriptor` de la table « `datatable` ». Dans le cas contraire, le nouveau descripteur est placé dans la table « `sd_table` » et l'index de la nouvelle ligne est utilisé.

**Structure des descripteurs de sécurité**

Dans le cadre du modèle discrétionnaire, le descripteur de sécurité comprend deux objets particulièrement critiques :

- le SID du propriétaire, qui dispose de privilèges implicites sur l'objet, se trouve en tête de structure ;
- la « *Discretionnary Access Control List* » (DACL) est la seconde partie qui nous intéresse dans notre analyse.

La DACL est une structure contenant un certain nombre « d'`Access Control Entries` » (ACE). Ces structures sont définies dans `WinNT.h` [3]. On peut distinguer ainsi ACE d'autorisation, ACE d'interdiction et ACE de journalisation.

Le propriétaire ne possède pas nécessairement d'entrée dans la DACL, mais il peut implicitement lire les permissions sur l'objet (« `READ_CONTROL` ») et les modifier (« `WRITE_DAC` »).

Le champ « `Trustee` » contient le SID d'un utilisateur ou d'un groupe titulaire d'une permission, à qui l'ACE s'applique.

Le champ « `AccessMask` » [5,6] est un bitmask contenant les différentes permissions d'une ACE donnée :

- droits génériques, visant à faciliter le portage des applications : lecture, écriture, exécution, tous ;

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Droits génériques | | | | Bits réservés | | | Accès SACL | Droits standards | | | | | | | | Droits spécifiques | | | | | | | | | | | | | | | |

**Figure 3.** Access Mask

– droits standards, applicables à tous les types d'objets : synchronisation pour accès exclusif, modification du propriétaire, modification du descripteur de sécurité, sa lecture, suppression de l'objet ;
– droits spécifiques, qui dépendent du type d'objet.

Notons le cas du bit n°8 (`0x100`) qui, lorsqu'il est positionné dans l'`AccessMask`, permet l'utilisation des droits étendus. Ceux-ci sont propres aux objets d'une application donnée et sont représentés par un numéro de série de type `GUID`. Deux champs supplémentaires peuvent être présents pour une telle ACE :

– le numéro GUID du droit en question : `Object-Type` ;
– le numéro GUID du type d'objet auquel peut s'appliquer le droit, ou apte à en hériter le cas échéant : `Inherited-Object-Type`.

Si aucun GUID n'est renseigné dans l'`Object-Type`, tous les droits étendus sont accordés au titulaire.

Ces champs supplémentaires peuvent également être présents pour une ACE « normale » : l'`Object-Type` permet alors de restreindre la portée de l'ACE à un attribut unique d'un objet. Son absence ne limite pas l'ACE à un attribut donné.

Les GUID de droits utilisés pour le champ `Object-Type` sont stockés sous forme de chaîne de caractères dans le conteneur Extended-Rights de la partition de Configuration de l'Active Directory. La colonne correspondante est `ATTm590164`. Les GUID utilisés pour le champ `Inherited-Object-Type` sont, eux, attachés à chaque classe d'objet dans le schéma. Ils sont stockés dans la colonne `ATTk589972`, au format binaire défini dans le fichier d'en-tête `Guiddef.h` du SDK de Windows.

### Descripteurs de sécurité liés à Microsoft Exchange

Notons que dans le cas d'extensions du schéma, telles que celle opérée lors de l'installation de l'architecture de messagerie Exchange dans un environnement Active Directory existant, des attributs supplémentaires sont ajoutés. Il faut résoudre leur nom de colonne de la même manière, toutefois ces noms peuvent varier d'une extension de schéma à une autre, ou d'une version à l'autre. Un attribut est particulièrement intéressant : `ms-Exch-Mailbox-Security-Descriptor`. Il concerne la boîte aux lettres

de l'objet (utilisateur, groupe...), c'est celui qu'il faut récupérer pour auditer les permissions attachées aux boîtes aux lettres des utilisateurs. Il stocke un entier qui est un index vers la table « `sd_table` », aussi utilisée pour le descripteur de sécurité de l'objet lui-même. Cette table stocke de manière indifférenciée tous les descripteurs de sécurité, qu'ils soient utilisés par Active Directory ou par Exchange.

D'un point de vue pratique, le nom de colonne correspondant à l'attribut `ms-Exch-Mailbox-Security-Descriptor` n'est pas obtenu à partir d'`Attribute-ID` comme les attributs par défaut d'un annuaire Active Directory, mais avec « `msDS-IntId` » qui contient les noms de colonnes des attributs étendus.

## 5 Outillage pour l'audit

### 5.1 Approche

Il s'agit de présenter une vue des ACE interprétable par un humain afin de permettre le filtrage et le rejeu de requêtes prédéfinies par diverses heuristiques. Le but est également de faciliter l'analyse des données « à la volée », sans critères prédéfinis : l'auditeur doit pouvoir parcourir les objets et les permissions pour définir au fur et à mesure des critères de filtrage. Cette optique est extrêmement précieuse dans le cadre d'une analyse post-compromission pour adapter les critères de recherche à chaque cas et ne pas laisser passer des permissions douteuses au travers des mailles du filet.

Représenter la base de données des ACE sous la forme d'un schéma relationnel classique facilite grandement le développement d'outils ou d'heuristiques. Ceci facilite également l'analyse d'une base Active Directory depuis un simple ordinateur portable d'audit.

Il faut néanmoins garder à l'esprit que pour des entités de taille moyenne, le volume d'informations à traiter sera rapidement gigantesque (plusieurs dizaines de millions d'ACE), une fois les résolutions de liens effectuées.

Le prototype réalisé se base sur une simple base SQL afin d'identifier les points durs dans le processus et si besoin utiliser des entrepôts de données plus adaptés (on pensera notamment aux bases de données de type « NoSQL »). Cet outil permet d'inverser la logique de l'interface graphique fournie par Microsoft (qui ne permet la visualisation et l'édition d'ACE qu'objet par objet, c.f. figure 4).

De manière générale, la procédure suivie consiste en la réalisation des étapes suivantes :

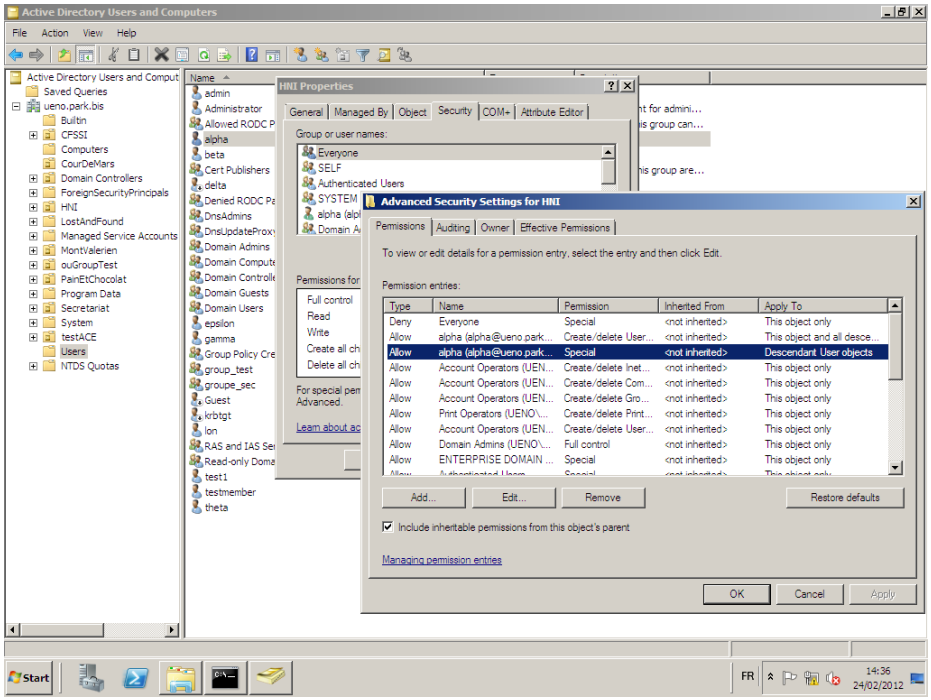



**Figure 4.** Affichage des permissions via l'interface graphique de Microsoft, objet par objet

- récupération d'une copie exploitable de la base ESE de l'Active Directory ;
- interprétation et export des données qu'elle contient vers un format intermédiaire ;
- import des données dans une base SQL ;
- résolution des liens entre objets, « *backlinks* », noms textuels des objets et interprétation des éléments numériques de type GUID, SID, etc. ;
- résolution des `Common-Name` et `DistinguishedName` des objets pour interpréter la signification des SID et repérer la position d'un objet dans l'arborescence ;
- précalcul des plus grosses jointures engendrant plusieurs dizaines de millions de résultats ;

Les sections suivantes présentent les résultats obtenus sur la base du prototype développé.

### 5.2 Difficultés principales

Certains points pénalisent fortement l'interprétation humaine des données contenues dans les bases ESE. Ainsi, les principales difficultés rencontrées au cours des expérimentations sont, entre autres :

- trouver les attributs intéressants, et surtout automatiser leur découverte à partir du schéma spécifique à chaque instance d'un Active Directory ;
- identifier toutes les entrées pertinentes à l'interprétation humaine permettant la résolution des données brutes sous forme de texte (GUID, `Common-Name` / `LDAP-Display-Name`...). Il s'agit ici de trouver les représentations textuelles recontrées dans l'interface graphique ;
- absorber la masse d'informations générées, qui pénalise grandement les performances. Il n'est pas rare d'avoir à auditer plusieurs millions, voire dizaines de millions d'ACE : les recherches sur les champs textuels, pourtant pratiques d'un point de vue humain, ne sont alors pas les plus efficaces et peuvent prendre plusieurs dizaines de secondes.

Ainsi, le traitement des bases ESE ne couvre pas l'ensemble des tables et données qu'elles contiennent. Autant que possible, les données sont conservées sous forme numérique. L'indexation de la base s'effectue sur les champs utilisés pour la majorité des recherches et augmente sensiblement les performances de l'ensemble. Cette indexation est effectuée sur les tables préliminaires représentant d'une part les ACE et d'autre part les descripteurs de sécurité. Puis, après jointure de celles-ci dans une table d'agrégat, ces champs sont à nouveau indexés.

Les champs utiles à l'indexation sont notamment les identifiants de descripteur de sécurité (`sd_id`, `NT-Security-Descriptor` et `ms-Exch-Mailbox-Security` les colonnes contenant des SID d'objet, de propriétaire ou de titulaire de droit, la colonne `AccessMask` ainsi que la colonne `Common-Name` pour rechercher les objets par leur nom.

#### Considérations de performances

La réalisation des opérations citées précédemment pose plusieurs problèmes de performance qui impactent plus ou moins directement les possibilités d'exploitation des données dans les différents cas d'audit évoqués.

Le parcours des données de la base ne doit pas être ralenti ; il est inacceptable d'avoir des requêtes de plus de quelques secondes, car cela pénalise fortement l'analyse « à la volée » et ne permet pas d'avoir la souplesse nécessaire dans des cadres particuliers, par exemple au cours

de la reconstruction d'un système d'information compromis. Dans ce cas de figure, l'audit complet doit pouvoir être itéré avec des cycles relativement courts (de l'ordre de quelques heures) pour suivre le déploiement des mesures correctrices dans le temps. De plus, pour les bases assez larges qui risquent d'engendrer une explosion combinatoire, il est préférable de commencer le parcours des entrées avant la fin de la résolution des liens et des jointures.

*Active Directory de test*

Dans le cadre de cet article, un Active Directory de test a été créé sous Windows 2008 R2 avec une installation de Microsoft Exchange 2010. Pour les besoins des tests et démonstrations, les groupes, délégations de privilèges, accès aux boîtes aux lettres ont été peuplés et paramétrés en suivant les recommandations et documentations fournies par Microsoft à destination des administrateurs de domaines.

Certains écarts ont été introduits volontairement dans le domaine :

- des mauvaises pratiques d'administration pour la gestion des privilèges ;
- des utilisateurs malveillants mettant en place divers moyens de maintenir certains privilèges.

Nous nous baserons sur ce domaine pour présenter notre démarche d'audit dans la section 6. Avant cela, on se propose d'étudier la base résultante d'un point de vue purement technique.

*Taille des bases de données engendrées*

Le domaine de test comprend un total de 14052 objets, dont les principaux sont répartis comme suit selon leur `Object-Category` :

- `User` : 6453 (les utilisateurs du domaine) ;
- `Attribute-Schema` : 3431 (chacun des attributs utilisables, extensions de schéma comprises) ;
- `Group` : 100 (groupes d'utilisateurs) ;
- `Organizational-Unit` : 11 (chacune des unités organisationnelles configurées dans le domaine) ;
- `Group-Policy-Container` : 4 (objets définissant les GPO applicables aux différentes OU).

Les tables des descripteurs de sécurité et d'ACE « *single instanced* » contiennent respectivement 14052 et 2214 entrées (pour 2,5 Mo et 432 Ko d'espace disque après extraction des bases ESE). Le premier nombre reflète évidemment le nombre de *securable objects* présents dans le domaine, tandis que le second révèle que beaucoup d'ACE sont partagées par les

objets : en effet, seuls 210 descripteurs de sécurité différents sont recensés. Beaucoup d'utilisateurs partagent des permissions identiques.

Le croisement des deux tables permet de résoudre l'ensemble des correspondances entre les objets et ACE qu'ils portent. Ceci résulte en 171767 entrées et leur représentation en base de données occupe 73,7 Mo. L'opération complète avec la résolution de toutes les références et des identifiants a duré une dizaine de minutes sur un processeur Intel X5650 (2.66 Ghz, hexacore).

Au cours d'audits précédents, les bases Active Directory étaient beaucoup plus volumineuses. Par exemple, un domaine comprenant environ 50000 comptes d'utilisateurs et environ autant de machines possédait 6,6 millions de descripteurs de sécurité, 15 millions d'ACE objet à auditer, plus 1,2 millions d'ACE concernant à elles seules les droits sur les objets Exchange. Pour être exploitables, ces bases ont été filtrées pour exclure certains types d'objets de l'analyse ; il restait tout de même environ 7 Go de données à analyser (les précalculs duraient environ une heure).

*Précalcul des jointures*

Les recherches de l'auditeur vont continuellement alterner entre, notamment, des filtres sur les objets concernés par une ACE (table des descripteurs de sécurité) et les titulaires de droits sur ceux-ci (table des ACE). Les volumes d'information concernés par ces croisements empêchent l'exploitation de la base dans des conditions correctes sur un portable d'audit, voire une simple machine dédiée : les requêtes sur l'exemple précédent prenaient jusqu'à 80 secondes pour terminer.

Les mécanismes de cache, tables temporaires ou vues sont fastidieux à configurer convenablement : leur consommation en espace disque explose rapidement (et ralentit fortement le système lors du chiffrement de l'espace de stockage). Les conditions de réalisation de l'audit empêchent l'exploitation de serveurs dédiés avec des quantités de mémoire vive suffisantes. Sur MySQL et PostgreSQL, des tables temporaires occupant jusqu'à 50 Go de données ont été observées en raison des multiples critères de recherche et de tri différents utilisés par les auditeurs.

Ces raisons poussent à précalculer les jointures requises au croisement des tables : une double jointure externe, pour résoudre les identifiants de descripteurs de sécurité puis pour résoudre les SID sous forme de texte (`Common-Name`). Si ceci participe à la création de tables « à plat », ce précalcul accélère sensiblement les opérations de recherche si tant est que les index soient correctement positionnés (voir la section 5.2). Les requêtes terminent alors en quelques secondes tout au plus.

*Résolution de valeurs « just in time »*

Cette préjointure ne suffit pas à l'interprétation des résultats : notamment, les GUID des champs `Object-Type` et `Inherited-Object-Type` ne sont pas résolus, de même que l'ensemble des champs contenant des SID ne le sont pas nécessairement (par exemple le *primary owner*).

C'est pourquoi ces champs sont résolus lors de la présentation des résultats par des sous-requêtes. Celles-ci auront lieu en nombre limité (on ne présente pas plus de quelques centaines de résultats à l'auditeur) et le temps de calcul est négligeable.

Cependant, la résolution « *just in time* » de leur signification limite les possibilités de tri et recherche « textuelle ».

En revanche, la technique permet également le décodage des structures de l'Active Directory pour assister l'auditeur dans leur interprétation, comme notamment l'`AccessMask`, les `AceFlags` [3] ou encore les `Object-Category` (dont une valeur peut correspondre à plusieurs descriptions équivalentes). Ce décodage intervient uniquement à l'affichage et permet par exemple d'éviter de surcharger les résultats par l'ensemble des valeurs d'un champ multi-valué.

### 5.3 Présentation de l'interface d'analyse

L'interface d'analyse présente les données représentant les ACE de manière tabulée (figure 5). On y retrouve le croisement des *security descriptors* avec les objets concernés, ce qui permet une vision des éléments réduite aux champs essentiels.

Intégrés dans la présentation des résultats, on retrouve des contrôles (`!=` et `==`) permettant la mise en place rapide de filtres qui modifient la requête exécutée. D'un clic, on peut ainsi exclure une valeur particulière d'un champ pour filtrer les résultats.

Elle permet également l'utilisation de filtres plus élaborés (figures 6 et 7) comme des comparaisons entre champs, la génération automatisée d'agrégats, ou encore la jointure avec d'autres tables de résultats, par exemple pour corréler les informations contenues dans la base des ACE avec des données provenant d'analyses indépendantes de l'Active Directory (corrélation des noms d'utilisateurs avec des journaux d'authentification réseau par exemple).

Enfin, certaines facilités sont intégrées pour assister l'auditeur et rappeler la logique de l'Active Directory. Le décodage automatique des structures binaires permet leur interprétation sans surcharger les lignes de résultats (figure 8) et le système permet la modification des descriptions

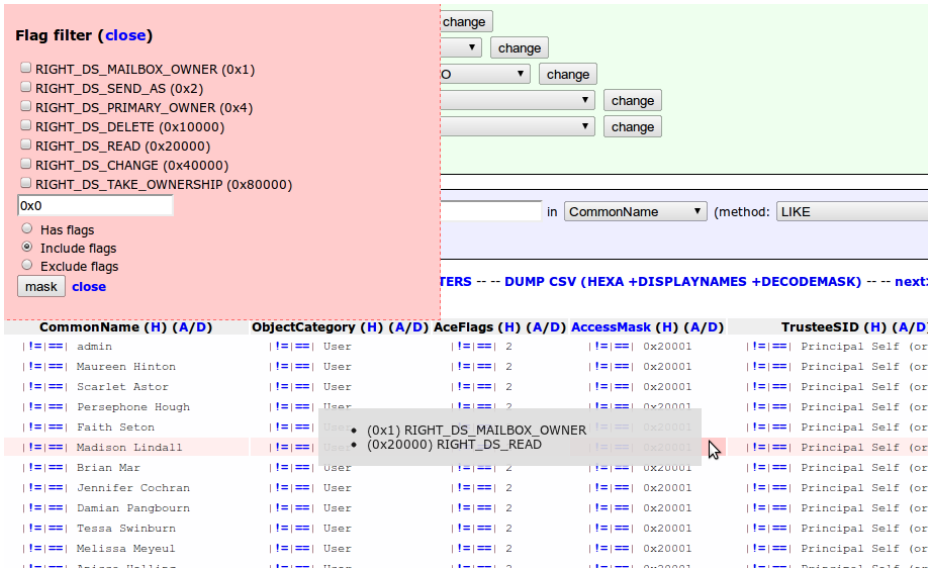



**Figure 5.** Vue générale de l'interface

textuelles de certains éléments dont la correspondance n'a pu être résolue. En effet, il est fastidieux de retrouver dans la base ESE l'ensemble des correspondances avec les GUID représentant des propriétés d'objets ou des actions réalisables. Cependant, il est possible pour l'auditeur de renseigner ces éléments au cas par cas et d'alimenter ainsi une base de connaissances commune qui sera réutilisable d'un audit à l'autre.

Les filtres générés par l'auditeur au fur et à mesure de son analyse sont regroupés dans une zone (figure 9) synthétisant leur équivalence en langage SQL et permettant facilement le retrait de tout ou partie des filtres.

Enfin, il est possible d'exporter et d'importer une liste de filtres, ainsi que de sauvegarder une session de recherche et de la partager avec un collaborateur pour mieux se répartir le travail d'audit. Cette fonctionnalité est également précieuse pour sauvegarder une certaine vue des objets audités, afin de procéder à l'extraction de listes à destination des administrateurs du système audité.

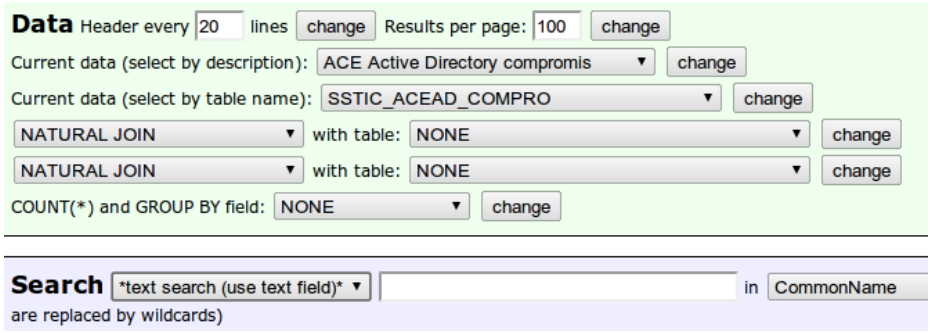



**Figure 6.** Définition de filtres, jointures et aggrégats


**Figure 7.** Définition de filtres sur des champs de bits, en fonction de la structure auditée

## 6 Méthodologie d'audit

### 6.1 Éléments attendus de l'audit des permissions

Cette section présente le déroulement d'un audit d'un Active Directory dans le but d'évaluer la sécurité apportée par les règles de contrôle d'accès aux objets qu'il contient. Cette méthodologie permet de révéler les discordances entre les règles voulues et configurées par l'administrateur et celles réellement présentes dans la base.

A l'issue de l'audit, l'approche présentée ici permettra de mettre en évidence deux cas principaux :

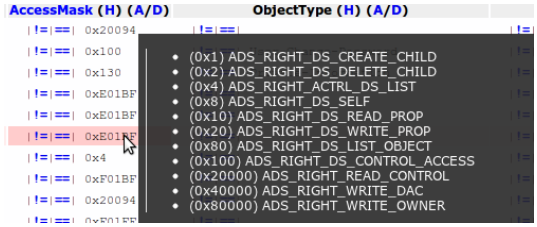



**Figure 8.** Décodage en surimpression du champ de bits « `AccessMask` »

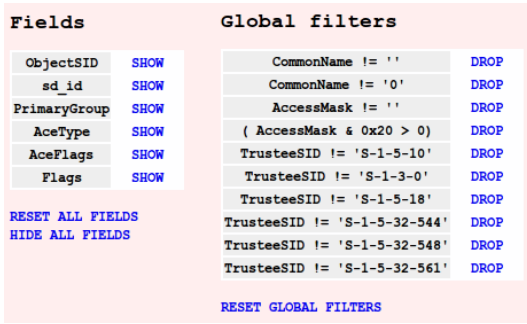



**Figure 9.** Liste des filtres actifs

– la présence d'écarts entre la situation voulue par l'administrateur pour la délégation de privilèges sur un sous-ensemble des objets et la situation réellement décrite par les ACE portées par les objets ;
– la présence d'ACE illégitimes sur certains objets, ou des droits excessifs d'un titulaire obtenus par les différents mécanismes d'héritage.

Le premier cas permet généralement de pointer les mauvaises pratiques d'administration ou la méconnaissance des mécanismes de délégation de privilèges proposés par l'éditeur.

Le second met en évidence, si ce n'est pas une maladresse, une tentative d'élévation ou de maintien de privilèges, ce qui peut révéler une compromission beaucoup plus étendue. On peut confirmer cette impression en cherchant si d'autres occurrences se retrouvent pour le même titulaire envers d'autres objets. C'est également l'occasion de communiquer les listes des titulaires de droits suspicieux en demandant la validation des privilèges observés par les responsables du système d'information.

On notera que l'approche préconisée par l'éditeur dans le cadre d'audits de sécurité d'un Active Directory (*security assessment*) ne met pas en œuvre l'audit exhaustif des ACE tel que présenté dans cet article, mais se concentre seulement sur quelques objets et les groupes par défaut.

La suite de cette section détaillera les points importants à auditer systématiquement puis nous donnerons une liste d'éléments de réflexion sur des critères de filtrage utiles. Nous rappellerons des objets particuliers de l'Active Directory et leurs subtilités. Enfin, il sera proposé un recensement et un résumé des points-clés à analyser dans les divers cas de figure évoqués (audit ponctuel ou réaction après incident).

### 6.2 Points-clés d'un audit de permissions Active Directory

Après avoir traité les bases ESE, une fois importées dans notre outil d'analyse, nous allons commencer leur analyse. Cette section déroule la méthode proposée et présente pour chaque point à couvrir des exemples de permissions anormales. Elle se veut être un « pense-bête » de l'auditeur, décrivant les points majeurs de la démarche proposée pour ne pas oublier les principales sources de problèmes de sécurité.

Malheureusement, les possibilités d'évasion sont multiples, tant les indirections dans la gestion des droits sont nombreuses. Les sections suivantes détaillent les principales possibilités de piégeage direct : les possibilités indirectes ne sont pas abordées, notamment celles permettant à un attaquant d'élever ses privilèges successivement ou par rebond.

#### Objets étudiés

Les objets intéressants d'un point de vue de la sécurité sont les suivants (listés par Object-Category) :

- `Computer` : les comptes des machines membres du domaine ;
- `User` (même catégorie que `Contact`, `Person`, `inetOrgPerson`, `Organizational-Pers`
  les utilisateurs du domaine, leur propriété `MemberOf` est virtuelle et calculée à la volée depuis les objets de type `Group` ;
- `Domain-Policy` : conteneurs de politiques du domaine, dont la *default domain policy* ;
- `Group` : les groupes d'utilisateurs (leur propriété `Member` définit les appartenances) ;
- `Group-Policy-Container` : il s'agit des conteneurs de GPO applicables aux OU du domaine ;
- `Organizational-Unit` : les définitions d'OU du domaine et délégations de privilèges dont elles sont l'objet ;
- `Server` : les définitions de contrôleurs de domaine reconnus ;
- `Trusted-Domain` : les liens d'approbation avec d'autres domaines.

Les privilèges d'écriture de propriété sur ces objets seront donc particulièrement intéressants à auditer, leur possession permettant de modifier

la configuration même du domaine à l'insu de ses administrateurs, voire de s'octroyer des droits particuliers.

**Propriétaires d'objets**

Les propriétaires d'objets possèdent des droits implicites sur ces derniers permettant le contrôle total de leur descripteur de sécurité (c.f. section 3.1).

Ceci permet de s'octroyer temporairement des droits particuliers sur ces objets du système sans qu'ils ne restent en permanence déclarés explicitement sous forme d'ACE : le propriétaire peut se les retirer sans se bloquer l'accès aux objets qu'il possède.

Généralement, les objets du domaine sont possédés par les groupes *builtin*, tels que `Domain Admins`, `Enterprise Admins`, `Administrators` ou encore l'utilisateur `Local System` (local à un contrôleur de domaine). Lorsque des objets sont créés par un utilisateur bénéficiant d'une délégation de privilège, le SID du titulaire (`TrusteeSID`) bénéficiant de l'ACE de délégation sur l'OU devient le SID du propriétaire (`primaryOwner`). Si l'utilisateur est membre d'un groupe « DSI » possédant la délégation de privilège, ce groupe sera le propriétaire de l'objet créé. En revanche, si l'ACE déclare directement l'utilisateur *beta* comme titulaire de la délégation, ce dernier deviendra propriétaire des objets créés, et le restera y compris après retrait de la délégation de privilèges.

Ce mécanisme permet de se maintenir des droits implicites sur des objets créés avant le retrait de délégations de privilèges.

[shown: 7 **start=0**]

| CommonName (H) (A/D) | PrimaryOwner (H) (A/D) | AccessMask (H) (A/D) | TrusteeSID (H) (A/D) |
|---|---|---|---|
| Communication | Parker Oxford | 0x10 | BUILTINWindows Authorization Access Group |
| Communication | Parker Oxford | 0x100 | Authenticated Users |
| Communication | Parker Oxford | 0x20094 | Principal Self (or Self) |
| Communication | Parker Oxford | 0x20094 | Authenticated Users |
| Communication | Parker Oxford | 0xF01FF | Domain Admins |
| Communication | Parker Oxford | 0xF01FF | Account Operators |
| Communication | Parker Oxford | 0xF01FF | Local System |

**Figure 10.** Utilisateur propriétaire de l'objet du groupe « *Communication* »

La figure 10 montre que l'utilisateur *Parker Oxford* est le propriétaire de l'objet représentant le groupe *Communication* sans être titulaire de droits envers celui-ci. Il peut néanmoins s'en rajouter et retirer à loisir.

**Propriétaires de boîtes aux lettres**

Les boîtes aux lettres Exchange n'ont pas, dans l'annuaire, de type d'objet qui leur soit propre, mais étendent le schéma du domaine. En effet, elles rajoutent une référence à un descripteur de sécurité spécifique pour chaque objet de type `User`.

Leur audit passe donc par l'analyse des objets `User`, mais dont le descripteur de sécurité concerné est spécifié dans la colonne `ms-Exch-Mailbox-Security-Des` (voir section 4.3).

Trois éléments principaux sont à considérer :

- les propriétaires de boîtes, reflétés par la présence de l'`AccessMask 0x1` (droit spécifique `RIGHT_DS_MAILBOX_OWNER`) ;
- les détenteurs de permissions de lecture sur les boîtes (utilisation du droit étendu `Receive-As`, présent dans la colonne `Object-Type`) ;
- les détenteurs de permissions d'envoi au nom d'un autre utilisateur (utilisation du droit étendu `Send-As`, présent dans la colonne `Object-Type`), qui permettent notamment d'usurper l'identité de quelqu'un et donner du crédit à une campagne de « filoutage ».

Les boîtes aux lettres de dirigeants peuvent légitimement être accédées par les secrétaires, mais cela est généralement restreint à quelques boîtes accessibles pour une personne donnée. Les groupes sont rarement utilisés pour représenter ces privilèges, et on accorde généralement la propriété de la boîte. La détention de privilèges sur de multiples boîtes révèle soit des oublis de la part des administrateurs, soit la présence d'un utilisateur bien curieux vis-à-vis des correspondances de ses collègues.

La figure 11 illustre bien ce cas d'utilisation : quelques utilisateurs possèdent des permissions pour l'accès à une boîte autre que la leur, tandis que *Indiana Dumfries* possède les permissions pour 8 boîtes différentes. On constate également la présence d'une boîte aux lettres accédée par différents groupes : *Challenge* est une boite accédée par les membres des groupes *Communication*, *Direction*, mais également par les deux utilisateurs *alpha* et *epsilon*. En l'occurrence, ce cas de figure peut être légitime [5].

Ces incohérences peuvent être rapidement mises en évidence par le décompte des boîtes accessibles par les différents titulaires. On détecte ainsi rapidement les excès (voir figure 12), ce qui ne serait pas possible par l'interface graphique de Windows.

**Titulaires de permissions d'écriture d'attributs**

La présence de l'`AccessMask 0x20` (`ADS_RIGHT_DS_WRITE_PROP`) dénote

5. Toute ressemblance avec ce qui serait caché dans un quelconque logo n'est pas forcément fortuite...

[shown: 17 **start=0**]

| CommonName (H) (A/D) | ObjectCategory (H) (A/D) | AccessMask (H) (A/D) | TrusteeSID (H) (A/D) | Trus |
|---|---|---|---|---|
| Challenge | User | 0x1 | alpha | al |
| Challenge | User | 0x1 | epsilon | ep |
| Andrea Holbech | User | 0x1 | Emily Buckbee | Em |
| Damian Artois | User | 0x1 | Victoria Haverill | Vi |
| Vanea Holland | User | 0x1 | Vicki Balfour | Vi |
| Landon Dole | User | 0x1 | Whitney Billings | Wh |
| Bailey Wassen | User | 0x1 | Sharon Clavering | Sh |
| Andrea Holbech | User | 0x1 | Indiana Dumfries | In |
| Bailey Wassen | User | 0x1 | Indiana Dumfries | In |
| Beatrice Nairne | User | 0x1 | Indiana Dumfries | In |
| Damian Artois | User | 0x1 | • (0x1) RIGHT_DS_MAILBOX_OWNER | |
| Landon Dole | User | 0x1 | Indiana Dumfries | In |
| Tamara Rue | User | 0x1 | Indiana Dumfries | In |
| Vanea Holland | User | 0x1 | Indiana Dumfries | In |
| Wayne Breckenridge | User | 0x1 | Indiana Dumfries | In |
| Challenge | User | 0x1 | Communication | Co |
| Challenge | User | 0x1 | Direction | Di |

**Figure 11.** Propriétaires de boîtes aux lettres Exchange

| TrusteeCN (H) (A/D) | count_TrusteeSID (A/D) |
|---|---|
| \|!=\|==\| Indiana Dumfries | \|!=\|==\| 8 |
| \|!=\|==\| Emily Buckbee | \|!=\|==\| 1 |
| \|!=\|==\| Sharon Clavering | \|!=\|==\| 1 |
| \|!=\|==\| alpha | \|!=\|==\| 1 |
| \|!=\|==\| epsilon | \|!=\|==\| 1 |
| \|!=\|==\| Communication | \|!=\|==\| 1 |
| \|!=\|==\| Direction | \|!=\|==\| 1 |
| \|!=\|==\| Victoria Haverill | \|!=\|==\| 1 |
| \|!=\|==\| Whitney Billings | \|!=\|==\| 1 |
| \|!=\|==\| Vicki Balfour | \|!=\|==\| 1 |

**Figure 12.** Nombre de boîtes aux lettres Exchange accessible comme propriétaire par un titulaire

de la possibilité d'écrire une propriété d'un objet. Ce droit peut être restreint à une propriété unique, ou global à toutes les propriétés de l'objet. Le mécanisme est décrit en section 4.3.

On cherche à savoir si des utilisateurs peuvent, par exemple, modifier de manière illégitime les appartenances à des groupes privilégiés du domaine ou bénéficiant de délégations de privilèges.

La figure 13 montre les titulaires de permissions d'écriture de propriétés. Le compte de *Devin Livingstone* a notamment des permissions d'écriture sur l'ensemble des propriétés du groupe *DSI-HNI*, ce qui lui permet d'en modifier les membres. De plus, *alpha* et *beta* ont la possibilité de réécrire le SID de l'utilisateur *testmember*, permettant à ce dernier de prendre n'importe quelle identité. Enfin, on remarque que l'utilisateur

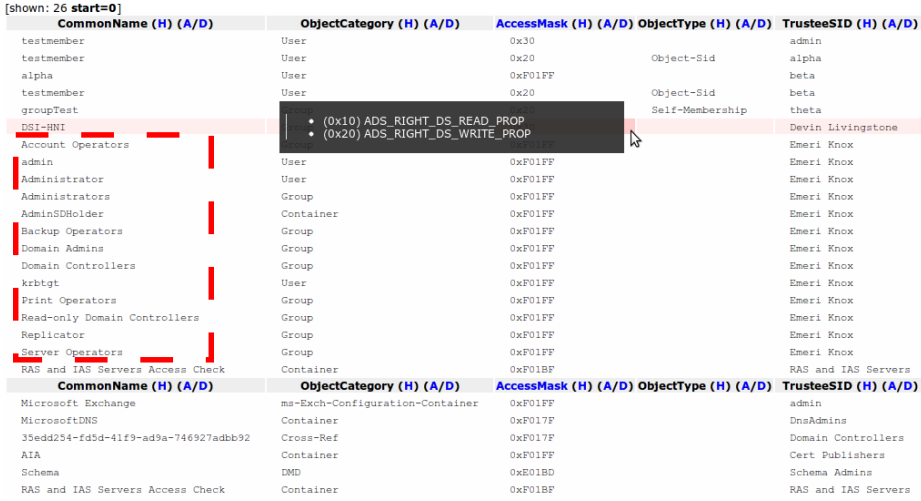



**FIGURE 13.** Titulaires de permissions d'écriture de propriétés d'objets

*Emeri Knox* possède le contrôle total des objets protégés du système (encadrés en rouge dans l'image).

### Titulaires de droits spécifiques

La détention de droits spécifiques est révélée par la présence de l'`AccessMask 0x100`. Si aucun GUID n'est précisé dans le champ `Object-Type`, toutes les actions étendues sont autorisées. En pratique, ce cas de figure est rarement rencontré de manière légitime pour des titulaires qui ne sont pas des comptes ou groupes d'administration par défaut, qui ont alors le contrôle total de l'objet (l'`AccessMask` est alors une constante incluant ce droit).

On portera une attention particulière aux bénéficiaires de droits tels que `User-Force-Change-Password`, permettant de forcer le mot de passe d'un utilisateur sans en connaître l'ancien, ou encore la détention de droits comme `DS-Replication-Get-Changes-All` permettant la réplication de tout ou partie de l'arborescence du domaine [6] au même titre qu'un contrôleur de domaine [7] (incluant donc les attributs protégés des utilisateurs, dont l'empreinte du mot de passe).

6. Ceci n'est possible que si l'utilisateur bénéficiant du droit `DS-Replication-Get-Changes-All` possède l'option `Server` dans son attribut `User-Account-Control`.

7. Lors d'une rump au SSTIC 2010, Aurélien Bordes présentait l'utilisation des RPC de réplication pour récupérer la base d'un contrôleur de domaine.

Les noms des droits spécifiques sont relativement explicites. Toutefois, dans le doute, il est préférable de se référer à la documentation Microsoft pour vérifier la portée de l'action concernée.

[shown: 8 **start=0**]

| CommonName (H) (A/D) | AccessMask (H) (A/D) | ObjectType (H) (A/D) | TrusteeSID (H) |
|---|---|---|---|
| Beatrice Nairne | 0x100 | User-Force-Change-Password | Devin Livingstone |
| Tamara Rue | 0x100 | User-Force-Change-Password | Devin Livingstone |
| Wayne Breckenridge | 0x100 | User-Force-Change-Password | Devin Livingstone |
| BISMAIL | 0x100 | Send-As | admin |
| BISMAIL | 0x100 | Receive-As | admin |
| Builtin | 0x100 | DS-Replication-Get-Changes-All | Domain Controllers |
| Server | 0x100 | SAM-Enumerate-Entire-Domain | BUILTINPre-Windows 2000 C |
| Builtin | 0x100 | Create-Inbound-Forest-Trust | BUILTINIncoming Forest Tr |

**Figure 14.** Titulaires de droits spécifiques

La figure 14 illustre les deux exemples précédents : l'utilisateur *Devin Livingstone* peut forcer le mot de passe de 3 autres utilisateurs à une valeur de son choix, tandis que les permissions de réplications du domaine sont possédées par le groupe des contrôleurs de domaine (ce qui est normal). Enfin, l'utilisateur *admin* peut envoyer et recevoir des messages au nom de la boîte aux lettres *BISMAIL*.

### Titulaires de délégations de privilèges sur les OU

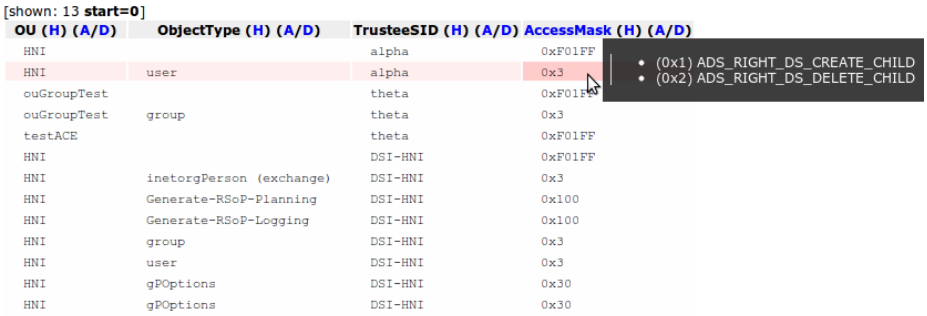
Les unités organisationnelles du domaine ne possèdent pas de `Common-Name`. Leur description textuelle se trouve dans la colonne `Organizational-Unit`. Ce sont les seuls objets du système pour lesquels elle est renseignée.

L'analyse des ACE portées par ces objets révèle les délégations d'administration accordées. De la même manière que n'importe quelle autre ACE, on retrouve potentiellement le type d'objet qu'elles concernent (par exemple, uniquement la manipulation d'utilisateurs) et les propriétés auxquelles elles sont restreintes le cas échéant (par exemple, uniquement l'appartenance à un groupe).

Dans le cas nominal, on trouvera en tant que titulaires de droits les groupes d'utilisateurs membres du service informatique local à une OU. En revanche, la présence d'utilisateurs explicitement déclarés comme titulaires de délégations dénote de mauvaises pratiques d'administration [8], ou de l'utilisation frauduleuse d'un compte à des fins malveillantes.

La figure 15 reflète d'une part les délégations normales accordées au groupe *DSI-HNI* sur l'OU *HNI*, et d'autre part deux ACE dont bénéficie

8. En effet, les bonnes pratiques recommandées par l'éditeur préconisent l'utilisation de groupes dédiés pour représenter les rôles d'administration [7].



[shown: 13 start=0]

| OU (H) (A/D) | ObjectType (H) (A/D) | TrusteeSID (H) (A/D) | AccessMask (H) (A/D) |
|---|---|---|---|
| HNI | | alpha | 0xF01FF |
| HNI | user | alpha | 0x3 |
| ouGroupTest | | theta | 0xF01FF |
| ouGroupTest | group | theta | 0x3 |
| testACE | | theta | 0xF01FF |
| HNI | | DSI-HNI | 0xF01FF |
| HNI | inetorgPerson (exchange) | DSI-HNI | 0x3 |
| HNI | Generate-RSoP-Planning | DSI-HNI | 0x100 |
| HNI | Generate-RSoP-Logging | DSI-HNI | 0x100 |
| HNI | group | DSI-HNI | 0x3 |
| HNI | user | DSI-HNI | 0x3 |
| HNI | gPOptions | DSI-HNI | 0x30 |
| HNI | gPOptions | DSI-HNI | 0x30 |

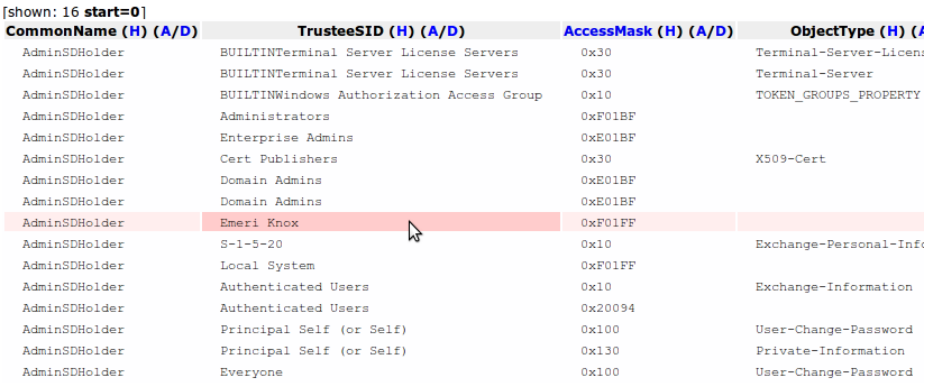


**Figure 15.** Titulaires de délégations de privilèges sur des OU

*alpha* sur l'OU *HNI* : il peut créer et supprimer des utilisateurs sous cette OU et bénéficie en plus du contrôle total (`AccessMask` à `0xF01FF`). L'utilisateur *theta* est dans un cas de figure similaire pour d'autres OU.

**Objets protégés du système : `adminSDholder`**

Le mécanisme spécifique lié à l'objet `adminSDholder` est décrit en section 3.2. Les ACE portées par cet objet font partie d'un descripteur de sécurité qui est également celui des objets protégés du système : la majorité des groupes privilégiés du domaine en font partie.



[shown: 16 start=0]

| CommonName (H) (A/D) | TrusteeSID (H) (A/D) | AccessMask (H) (A/D) | ObjectType (H) (A |
|---|---|---|---|
| AdminSDHolder | BUILTIN\Terminal Server License Servers | 0x30 | Terminal-Server-Licens |
| AdminSDHolder | BUILTIN\Terminal Server License Servers | 0x30 | Terminal-Server |
| AdminSDHolder | BUILTIN\Windows Authorization Access Group | 0x10 | TOKEN_GROUPS_PROPERTY |
| AdminSDHolder | Administrators | 0xF01BF | |
| AdminSDHolder | Enterprise Admins | 0xE01BF | |
| AdminSDHolder | Cert Publishers | 0x30 | X509-Cert |
| AdminSDHolder | Domain Admins | 0xE01BF | |
| AdminSDHolder | Domain Admins | 0xE01BF | |
| AdminSDHolder | Emeri Knox | 0xF01FF | |
| AdminSDHolder | S-1-5-20 | 0x10 | Exchange-Personal-Inf |
| AdminSDHolder | Local System | 0xF01FF | |
| AdminSDHolder | Authenticated Users | 0x10 | Exchange-Information |
| AdminSDHolder | Authenticated Users | 0x20094 | |
| AdminSDHolder | Principal Self (or Self) | 0x100 | User-Change-Password |
| AdminSDHolder | Principal Self (or Self) | 0x130 | Private-Information |
| AdminSDHolder | Everyone | 0x100 | User-Change-Password |

**Figure 16.** ACE portées par l'objet `adminSDholder`

Quelques ACE portées par l'objet `adminSDholder` sont illustrées par la figure 16. La plupart sont présentes de manière légitime, sauf une qui

donne les permissions de contrôle total pour l'utilisateur *Emeri Knox*. Ainsi, les groupes protégés du système deviendront les sujets de cette ACE lorsque le processus `SDPROP` l'y aura recopiée, ce qui explique pourquoi *Emeri Knox* est titulaire des ACE encadrées dans la figure 13.

## 7 Perspectives et conclusion

### 7.1 Sur la nécessité d'aller rapidement à l'essentiel

Certains audits concernent des bases de données gigantesques, comprenant plusieurs millions d'ACE à passer en revue. Pour permettre un traitement humain de cette masse d'informations, plusieurs techniques peuvent accélérer le travail.

L'auditeur voulant être exhaustif devra prendre plusieurs heures à quelques jours à filtrer, trier, exclure des ACE. De même, il devra se documenter sur l'ensemble des permissions observées pour justifier chacune des ACE. Quelques critères de recherche et de filtrage simples viennent alors à l'esprit :

- ne s'intéresser qu'aux objets principaux de l'Active Directory ;
- compter les permissions de chaque titulaire et étudier ceux qui en présentent un nombre conséquent, ou à l'inverse les personnes identifiées en possédant trop peu ;
- éliminer les utilisateurs appartenant à des groupes particuliers (secrétariats ou groupes *builtin* par exemple) après justification de l'entité auditée ;
- éliminer les utilisateurs légitimes tels que le propriétaire de sa propre boîte aux lettres [9] ;
- éliminer les permissions inintéressantes ou n'ayant pas d'impact important : on peut jouer sur l'`AccessMask` et l'octroi ou non de droits étendus, ou la restriction à des propriétés spécifiques (champs `Object-Type`, `Inherited-Object-Type`).

Ces critères peuvent être utilisés pour filtrer les ACE importées dans la base, et soulager ainsi le SGBD. La tâche qui s'annonçait titanesque avant l'application de filtres devient humainement réalisable dans des délais acceptables. Cependant, exclure des résultats de manière automatique pose le problème des faux négatifs : l'approche exhaustive reste préconisée pour ne pas laisser place au doute.

A titre d'exemple, le cas d'audit cité précédemment comprenait environ 1,2 millions d'ACE liées à Exchange. Après filtrage des cas légitimes

9. Filtres `Common-Name != TrusteeCN`, `ObjectSID != TrusteeSID` ou `TrusteeSID` ne contenant pas le SID `S-1-5-10`, `Principal Self (or Self)`.

ou sans conséquence pour la sécurité, l'audit des boîtes aux lettres ne produisait que quelques milliers de résultats qui, *in fine*, ne concernaient plus que quelques centaines d'utilisateurs. Le croisement avec les statistiques de répartition des ACE permettait alors de mettre rapidement en évidence les anomalies ou les comptes disposant de permissions excessives.

Il n'est pas rare de rencontrer des curiosités dans diverses configurations : suite à des migrations, on peut trouver des résidus sous la forme de groupes vides conservant des ACE pour des comptes désactivés ou inexistants, ou encore des propriétés d'objets possédant des valeurs impossibles. Un cas observé présentait des attributs `Members` et `MemberOf` incohérents, que l'interface graphique ne permettait pas d'obtenir manuellement. Un autre exemple était la présence d'ACE sur l'attribut virtuel `Membership` d'un utilisateur, chose qui n'était pas possible de réaliser par les outils d'administration standard, mais qui était sans conséquences : le système refusait les modifications de valeurs de cet attribut.

Cependant, l'analyse de ces cas de figure est relativement longue. Elle ne doit être faite que si l'on souhaite absolument justifier les ACE de chaque objet et elle révèle rarement une anomalie ayant un impact sur la sécurité.

### 7.2 La propriété adminCount

Nous n'avons traité que le problème des permissions accordées sur des objets de l'Active Directory. Tout ce qui concerne leurs attributs a été évité. Ils sont pourtant l'objet de mécanismes particuliers qui influent sur la sécurité de l'ensemble.

Une attention particulière peut être portée sur l'attribut « `adminCount` », qui est une autre forme de protection d'objets de l'Active Directory. Cet attribut est positionné sur un compte utilisateur dès qu'il devient membre d'un groupe protégé par l'objet « adminSDholder » (voir section 4.3). Lorsqu'un objet le porte avec la valeur 1, l'héritage des ACE est bloqué implicitement. L'attribut n'est plus retiré par la suite, même si l'utilisateur change de groupe.

Ce cas de figure permet à un compte de conserver les ACE qu'il porte directement et pour ne pas être affecté par les ACE héritées de l'unité d'organisation à laquelle il appartient. Ainsi, un utilisateur possédant cette propriété sera difficile à manipuler : il peut restreindre sa visibilité vis-à-vis des administrateurs délégués de son unité d'organisation ou se prémunir de restrictions de droits futures.

L'attribut est d'autant plus subtil qu'il n'est pas possible de l'observer via l'interface graphique standard disponible jusqu'à la version 2003 de

Windows (figure 17). A partir de la version 2008 du système d'exploitation de Microsoft, l'interface possède l'éditeur d'attributs qui révèle la propriété `adminCount` (figure 18).

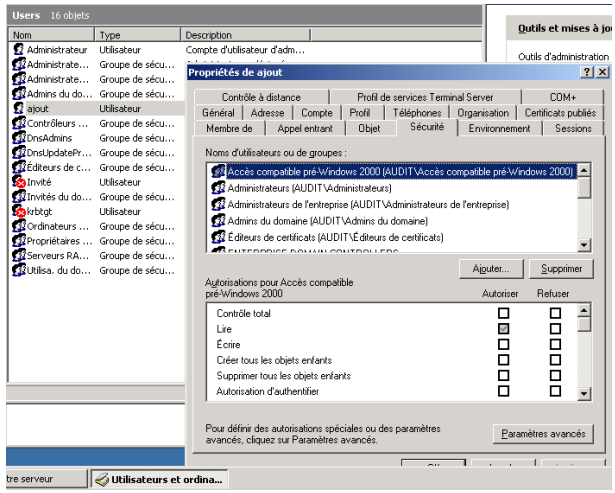



**Figure 17.** L'interface de Windows Server 2003 ne permet pas de visualiser les propriétés des objets de l'AD

Lorsqu'un administrateur modifie des permissions qui s'appliqueront soit directement, soit par héritage à un utilisateur qui possède la propriété `adminCount`, l'interface ne le prévient pas des comptes qui échapperont à ces modifications [10].

## 7.3 Conclusion et pistes de réflexion

L'approche présentée dans cet article et l'outil développé ont évolué en bénéficiant des retours d'expérience de chaque audit. Nous menons actuellement plusieurs réflexions pour améliorer la méthode et assister l'auditeur au maximum en le délestant de tâches inutiles.

Ainsi, les points suivants, envisagés suite au prototypage déjà effectué, sont laissés à l'appréciation du lecteur pour imaginer des évolutions :

10. L'équipe Directory Services de Microsoft précise sur son blog que la propriété n'est pas supprimée d'un compte dont on révoque des privilèges, en raison des résultats d'un sondage portant sur les pratiques d'administration (voir : http://blogs.technet.com/b/askds/archive/2009/05/07/five-common-questions-about-adminsdholder-and-sdprop.aspx).

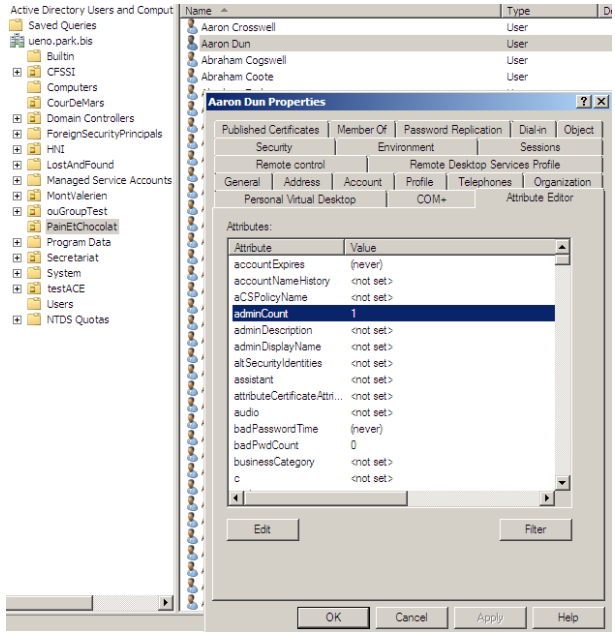



**Figure 18.** L'éditeur d'attribut disponible depuis Windows Server 2008 montre bien les propriétés d'objets

- développer une plate-forme d'analyse dédiée capable d'absorber la masse de données à traiter (avec des outils de type *hadoop*). Cependant, il faudra bien mesurer l'impact sur la réactivité des auditeurs, du fait de la nécessité d'un fonctionnement en « lots » de données à traiter ;
- permettre le travail collaboratif à plusieurs auditeurs, qui est améliorable par la création automatisée de sessions de recherche suivant différents critères de filtrage ;
- proposer un référentiel des ACE présentes dans une configuration standard d'Active Directory, selon les bonnes pratiques recommandées par l'éditeur, et n'afficher que les écarts avec ce référentiel ;
- systématiser le décodage automatique d'informations et le baser sur les structures tirées du SDK. Ceci, en plus de proposer à l'auditeur une aide contextuelle riche sur la signification des attributs étendus, leur impact et les points à ne pas omettre, permettrait d'auditer un système sans devoir en connaître (presque par cœur) toutes les subtilités ;

- proposer des méthodes et des exemples de représentations pertinentes des informations du contrôle d'accès (synthèses, tables, graphes, etc.) : les premières phases d'audit suivant l'import en base de données relationnelle pourraient comprendre la génération de graphes, basés sur les statistiques mentionnées dans les sections précédentes (par exemple, représenter graphiquement la répartition des ACE dont bénéficient des titulaires sur des boîtes aux lettres) ;
- étudier les possibilités de corrélation des données issues d'autres sources avec les résultats de l'audit des ACE : journaux d'authentification d'utilisateurs, groupes d'administrateurs d'application ou de postes, traces réseau. Les recherches peuvent être orientées à partir de comportements jugés anormaux. Notamment, un système de marquage dans la base de données permet de repérer un même compte utilisateur dans le contexte de l'audit des ACE et dans celui de l'audit des traces réseau ;
- étudier et traiter conjointement avec l'Active Directory les ACE stockées dans les descripteurs de sécurité du système de fichier NTFS, notamment pour les serveurs de fichiers, dont l'interprétation sera sensiblement identique à celle décrite dans cet article ;
- collecter et auditer les propriétés des objets de l'Active Directory pour tagger des objets sortant de l'ordinaire : attributs `adminCount`, drapeaux `UserAccountControl`, etc.

## Références


1. Thibaut Leveslin. Extraction des empreintes de mots de passe en environnement Windows. http://www.hsc.fr/ressources/presentations/gsdays2011_empreintes/index.html.fr, 2011.
2. Joachim Metz. Extensible storage engine (ese) database file (edb) format specification. http://sourceforge.net/projects/libesedb/files/Documentation/Extensible%20Storage%20Engine%20(ESE)%20Database%20File%20(EDB)%20format/, 2009.
3. Microsoft. *Access Control Entries*. http://technet.microsoft.com/en-us/library/cc961995.aspx.
4. Microsoft. *Access Rights and Access Masks*. http://msdn.microsoft.com/en-us/library/windows/desktop/aa374902(v=vs.85).aspx.
5. Microsoft. *ACCESS_MASK*. http://msdn.microsoft.com/en-us/library/windows/desktop/aa374892(v=vs.85).aspx.
6. Microsoft. *ADS_RIGHTS_ENUM enumeration*. http://msdn.microsoft.com/en-us/library/windows/desktop/aa772285(v=vs.85).aspx.
7. Microsoft. *Best Practices for Delegating Active Directory Administration*. http://www.microsoft.com/download/en/details.aspx?id=21678.
8. Microsoft. *Checking Control Access Right-Based Access*. http://msdn.microsoft.com/en-us/library/cc223518(v=prot.10).aspx.
9. Microsoft. *Description and Update of the Active Directory AdminSDHolder Object*. http://support.microsoft.com/kb/232199.
10. Microsoft. *DS-Heuristics attribute*. http://msdn.microsoft.com/en-us/library/windows/desktop/ms675656(v=vs.85).aspx.
11. Microsoft. *Protected Objects*. http://msdn.microsoft.com/en-us/library/dd240058(v=prot.13).aspx.
12. Microsoft. *Well-Known Security Identifiers*. http://technet.microsoft.com/en-us/library/cc978401.aspx.
13. Microsoft. *Well-known security identifiers in Windows operating systems*. http://support.microsoft.com/kb/243330.