Détecter DCShadow, impossible ?

Archived: 2026-04-05 18:49:33 UTC

Bonjour à tous,

Je vous propose aujourd'hui un article sur l'attaque présentée par Vincent Le Toux (@mysmartlogon) et Benjamin Delpy (@gentilkiwi) durant la conférence de sécurité BluehatIL 2018 qui a eu lieu les 23 et 24 janvier.

Leur présentation s'intitulait **Active Directory: What can make your million dollar SIEM go blind?**

En effet, l'intérêt pour un attaquant d'utiliser DCSHadow est de ne laisser aucune tracer de ses modifications puisque celles-ci sont effectuées sur une machine compromise par l'attaquant. Ainsi aucun log de modification AD n'est remonté.

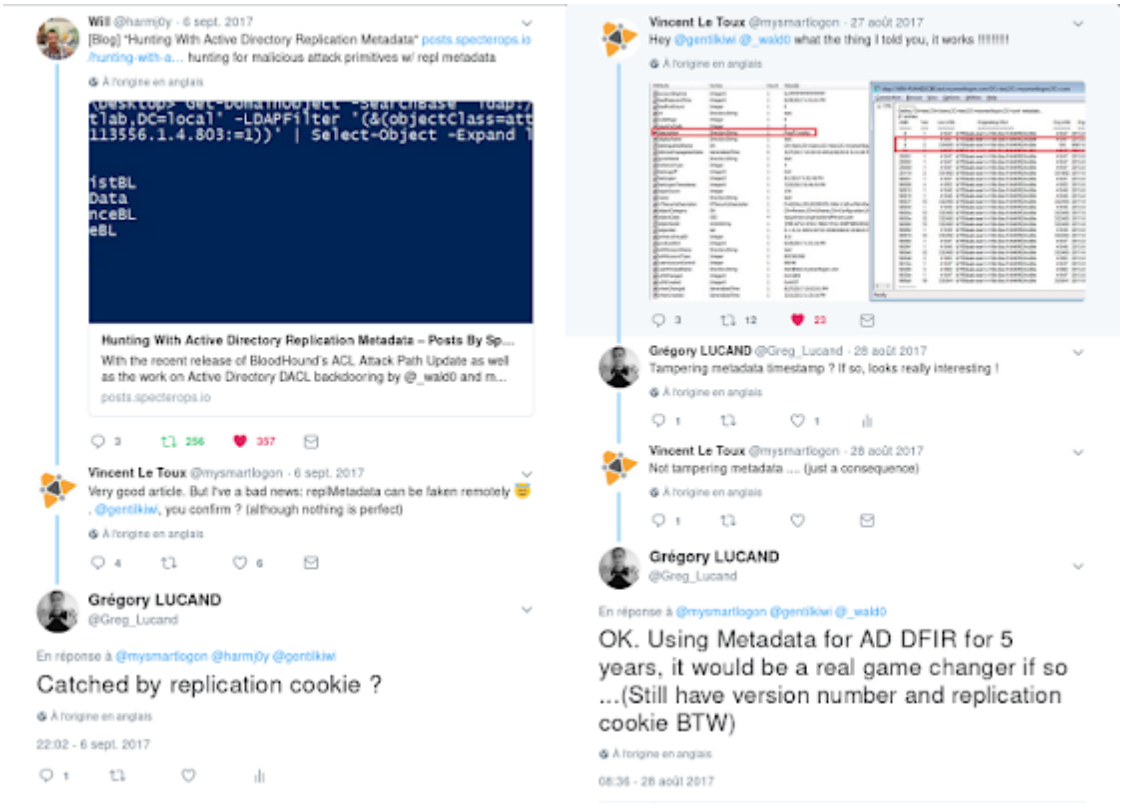
Vous pouvez retrouver la vidéo et les slides de la présentation sur le site officiel sur DCShadow https://www.dcshadow.com/.

Un peu d'histoire

La première évocation par Vincent et Benjamin de ce qui allait devenir DCShadow remonte à l'été 2017.

Faisant de la réponse à incident sur Active Directory depuis quelques années le tweet de Vincent m'a forcémment interpelé. La possibilité qu'un attaquant puisse altérer les métadonnées de réplication a toujours été une de nos craintes sur le forensic AD.

J'ai pourtant évoqué en répondant à ses tweets une autre méthode, non forensic mais plus de blue team, les cookies de réplication AD.



Les cookies de réplication AD sont assez méconnus mais utilisés par les équipes de réponse à incident AD lors des remédiations/reconstructions d'environnement Active Directory compromis.

Ils permettent notamment de limiter l'interruption de service lors de la remédiation contrairement aux précédentes méthodologies.

C'est un sujet assez peu documenté mais on peut en trouver une trace dans la présentation de l'ANSSI sur TV5 Monde du SSTIC 2017 Retour technique de l'incident de TV5Monde.

Cookie de réplication AD

Un cookie de réplication est un fichier généré par l'utilisation d'une extension de serveur LDAP, le contrôle DirSync. Il faut d'abord initialiser le cookie pour pouvoir avoir les modifications depuis la dernière utilisation du cookie.

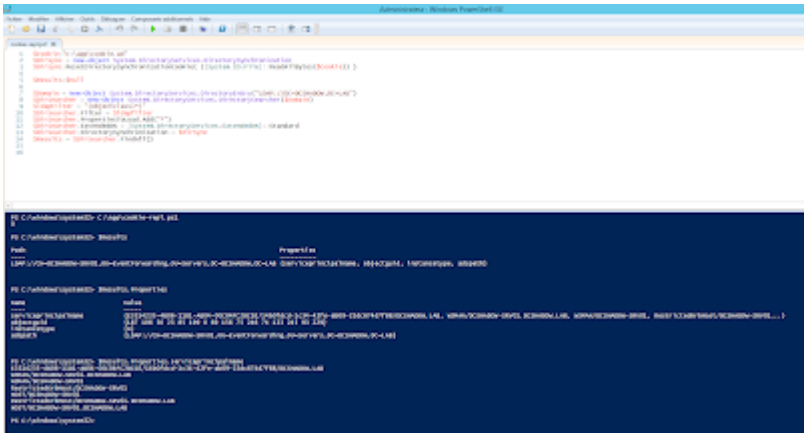
Vous retrouverez dedans une page dédiée au contrôle DirSync.

Pour utiliser ce contrôle, il ya plusieurs possibilités.

La plus simple pour tester rapidement, c'est d'utiliser Repadmin avec le switch /showchanges (visible avec l'aide en mode expert de repadmin /experthelp).

C'est ce que je vais utiliser dans cet article.

Ca marche très bien avec du powershell :

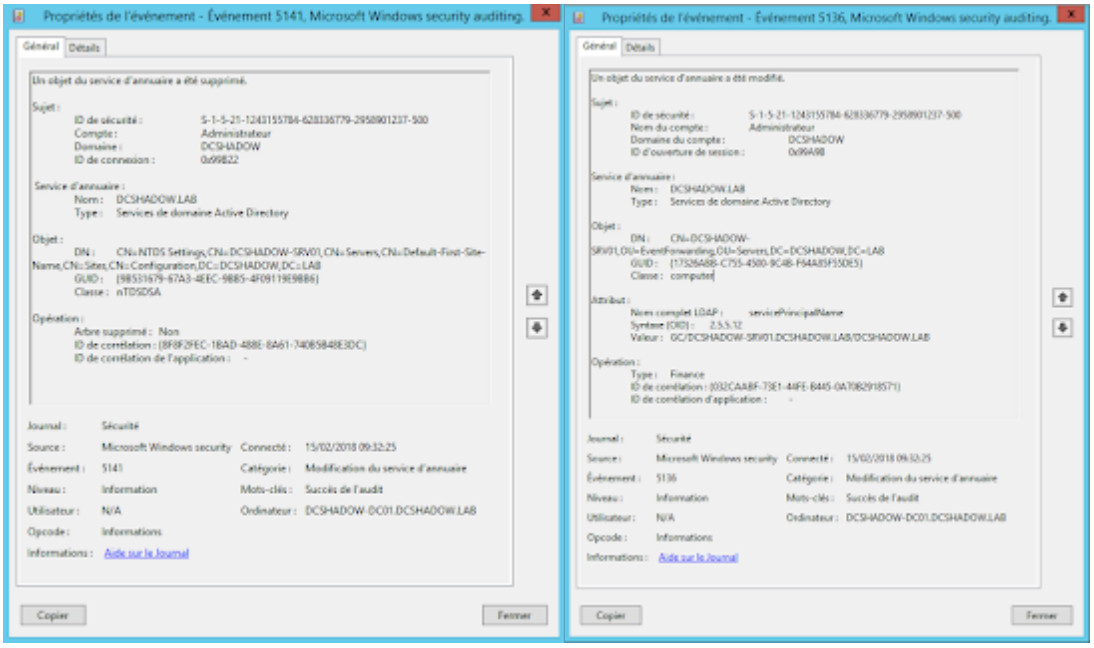


Je ne vais pas m'étender c'est relativement simple à utiliser. Revenons maintenant sur l'attaque DCShadow.

Détection de l'utilisation de DCShadow

Détecter l'utilisation de DCShadow peut paraître assez trivial. En effet, pour être utilisé, DCShadow doit au préalable effectuer des modifications sur des objets Active Directory. En journalisant la modification de ces objets, on peut détecter facilement l'utilisation de DCShadow.

Il suffit pour cela de configurer correctement la politique d'audit (audit réplication détaillée et modifications AD) et de configurer quelques SACL (écriture de l'attribut servicePrincipalName des objets Computer).



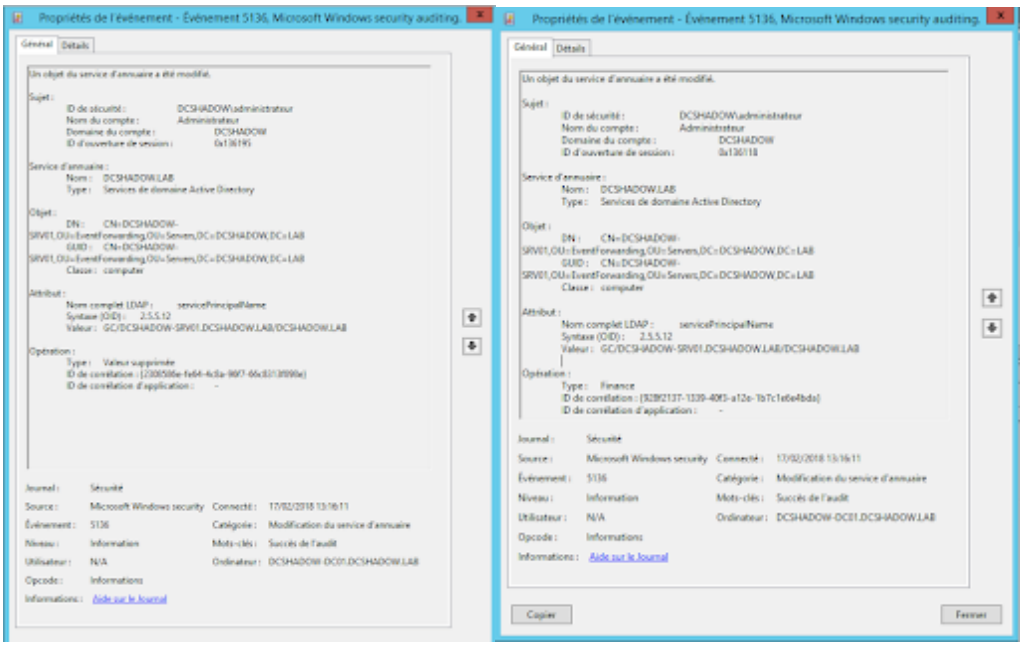
Il ya toutefois un gros bémol à cette technique. L'attaquant en capacité d'utiliser DCShadow dispose de privilèges élevés sur l'Active Directory. Rien ne l'empêche de modifier temporairement la politique d'audit sur le contrôleur de domaine qui sera ciblé par DCShadow pour la réactiver par la suite.

Ca nous laisse donc la détection par le réseau qui est beaucoup moins trivial à mettre en place ou les cookies de réplication.

Il est intéressant d'ailleurs de comparer ce que remonte la journalisation AD avec ce que remonte le cookie de réplication.

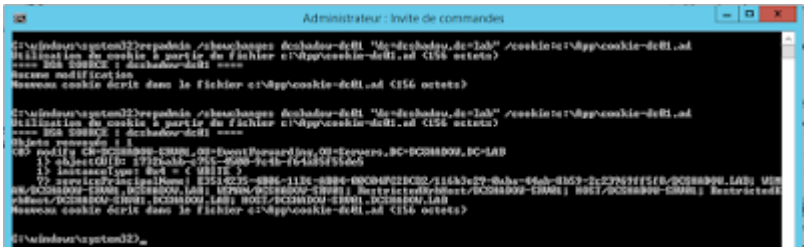
J'effectue une modification avec DCShadow.

Voici ce que j'ai dans le log:



On peut voir l'ajout puis la suppression du SPN de catalogue global (GC/).

Voici ce qu'on a avec le cookie de réplication:



Avec le cookie de réplication, on a la valeur finale du SPN, avec le GUID du service DRS mais pas le GC qui a été supprimé.

Ce qui est intéressant c'est que la journalisation ne log pas l'ajout du SPN du service DRS (GUID E35...).

On peut donc déjà détecter l'utilisation de DCShadow sans utiliser la journalisation ni la capture réseau.

Comme je l'ai déjà dit, l'intérêt de DCShadow est de ne pas laisser de trace sur les modifications en elles-même.

C'est bien de détecter que DCShadow a été utilisé mais ne pas savoir ce qui a été modifié rend la remédiation imposible.

On va voir qu'il y a plusieurs cas en fonction de comment on utilise DCShadow.

Mais avant de voir tout ça nous allons faire un très court rappel sur ce qui déclenche une réplication entre les contrôleurs de domaine car ça a énormément d'importance pour la suite.

Réplication AD et USN

La réplication AD se base sur le numéro USN (Update Sequence Number).

Pour faire très simple, le DC qui demande les réplications à son partenaire demande les modifications effectuées depuis le dernier USN qu'il connait.

Autrement dit, ton dernier USN que je connais est X, envoie-moi les attributs qui ont un numéro USN supérieur.

Pour plus d'informations, vous pouvez vous référrez à cet article du technet : [Tracking updates]

Comme je l'ai dit l'USN a beaucoup d'importance pour la suite.

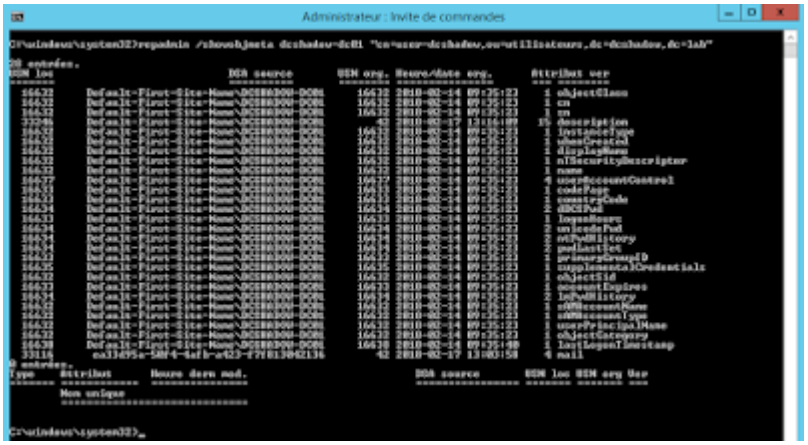
Détection des modifications opérées via DCShadow

Je n'ai pas pu tester tous les cas possibles en quelques heures mais voici déjà un bon aperçu.

Modification d'un attribut existant sur le DC où a eu lieu la dernière modification de l'attribut avec les switchs de DCShadow par défaut

Le premier cas que l'on va voir est l'utilisation de DCShadow telle qu'elle a été faite au début de l'article.

Si l'on regarde les métadonnées de l'objet nous avons ceci :



On peut voir que la valeur du numéro USN de l'attribut Description est 42, nombre assez remarquable.

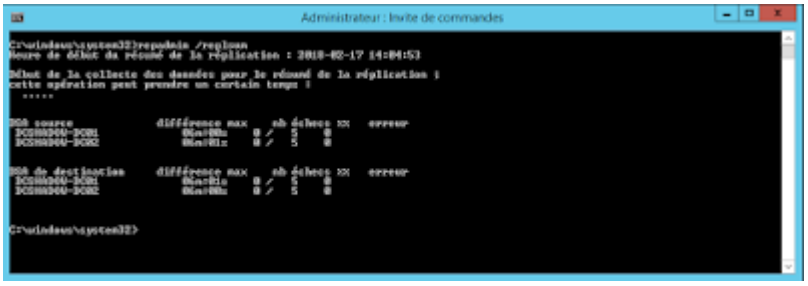
En regardant le code source, on trouve rapidement que c'est une valeur hardcodée et qu'il ya une branche conditionnelle (switch qu'on verra plus tard).

```
1724                    if(object->pAttributes[i].MetaData.dwFlag & REPLICATION_USN_SET)
1725                        (*ppReplEnt)->pMetaDataExt->rgMetaData[i].usnOriginating = object->pAttributes[i].MetaD
1726                    else
1727                        (*ppReplEnt)->pMetaDataExt->rgMetaData[i].usnOriginating = 42;
```

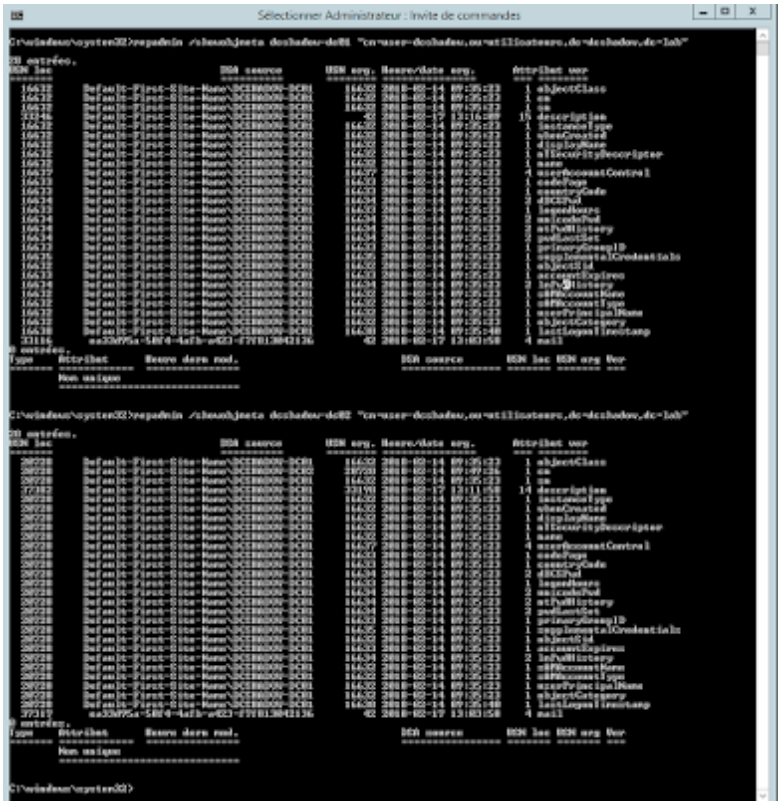
Ce qui m'a étonné quand j'ai joué la première fois avec DCShadow, c'était que cette modification n'était pas détectée par le cookie de réplication.

Mon premier réflexe a été de me dire qu'il sera catché par le cookie sur mon second DC lors de la réplication, mais je n'avais rien non plus et surtout l'attribut n'était pas modifié sur le second DC.

Je vérifie les réplications et il n'y a pas de problème.



Je compare ensuite les métadonnées de réplication.



Et là on peut voir une différence entre les 2.
Ma modification effectuée avec DCShadow n'a donc pas été repliquée sur les autres contrôleurs de domaine.

La raison est simple, la valeur d'USN étant 42, la modification n'est pas répliquée.

On voit tout de suite un intérêt du point de vue DFIR.

Premièrement, 42 est un marquant fort dans les métadonnées de réplication (mais peut-être modifié dans le code par l'attaquant).
Deuxièmement, les modifications n'ayant pas été repliquées, la remédiation des modifications est très simple, il suffit de reconstruire le DC (après avoir réglé le problème de privilège bien évidemment).

Dans ce cas, on ne se sert pas du cookie de réplication puisqu'il n'y a pas réplication mais on peut retrouver les modifications via les métadonnées (ou comparer les données entre le DC sur lequel DCShadow a été utilisé et un autre DC).

L'intérêt pour l'attaquant est bien évidemment de ne pas utiliser DCShadow dans ce sens. Puisque si on détecte l'utilisation de DCShadow sur ce contrôleur de domaine on retrouvera rapidement les modifications qu'il a effectuées.

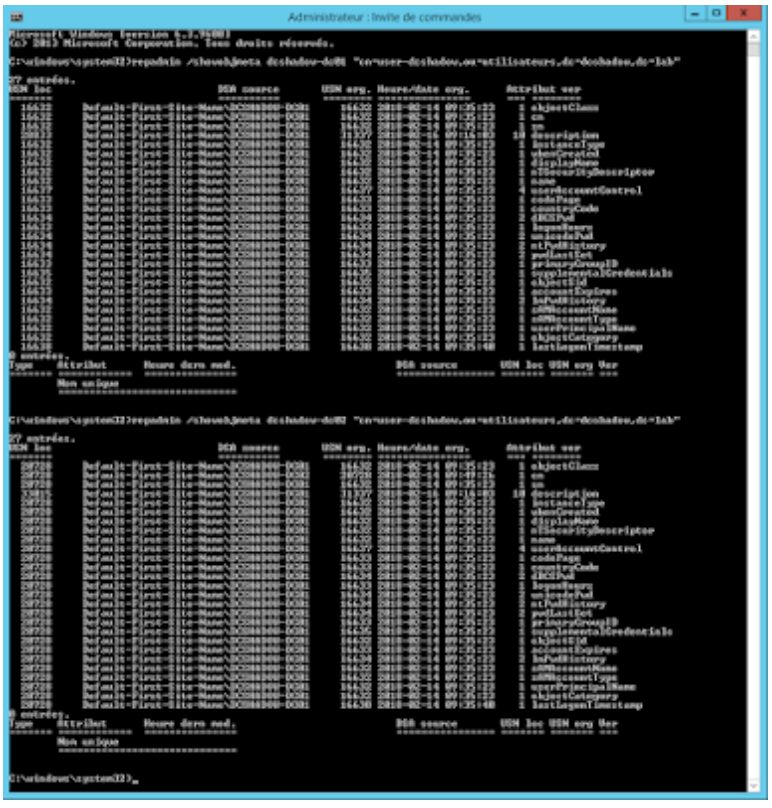
On va donc utiliser l'autre branche conditionnelle du code qui est un switch permettant de spécifier le numéro USN.

Modification d'un attribut existant sur le DC où a eu lieu la dernière modification de l'attribut en utilisant le switch replOriginatingUsn de DCShadow

Cette fois, nous allons spécifier la valeur du numéro USN.

On recommence cette fois en utilisant le switch /replOriginatingUsn avec un numéro USN permettant la réplication.

On vérifie les réplications.



Et là, on voit que la modification de l'attribut Description a bien été répliquée.

D'un point de vue DFIR c'est la merde, les modifications ont été répliquées et je n'ai plus le marquant du numéro USN.
Bonne chance pour remédier ...

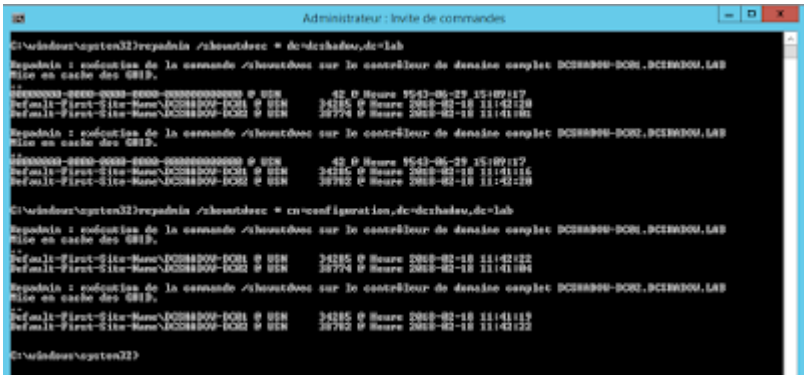
Mais cette fois, si on regarde le cookie de réplication:

On voit bien la modification avec la valeur de l'attribut modifié.
Il est assez facile de retrouver les modifications opérées avec DCShadow puisqu'elles vont suivre la modification du SPN.

Autre marquant

Je me suis demandé si l'utilisation de DCShadow pouvait provoquer un USN Rollback.

En réflechissant, ça ne peut pas être le cas mais en regardant la valeur de Up-to-dateness Vector j'ai eu la surprise de voir ceci.



A creuser mais ça pourrait être un marquant de l'utilisation de DCShadow si on n'a pas déployé de moyen de détection.

Conclusion

On peut donc désormais répondre à la question de cet article, est-ce vraiment impossible de détecter DCShadow ?

Il faut bien séparer utilisation de DCShadow et modifications opérées par DCShadow.

Dans les 2 cas on a pu voir qu'il était possible de les détecter (Bien que je n'ai pas pu tester tous les cas possibles). Cette détection s'avère tout de même très difficile.

Clairement si vous n'êtes pas en capacité d'empêcher l'attaquant de récupérer les privilèges nécessaires à l'utilisation de DCShadow, il est peu probable que vous soyez en capacité de déployer une détection basée sur les cookies de réplication.
