Délégation Kerberos

Workshop GCC - Mai 2026

Zleb & Cqban

Objectifs du workshop

Objectifs du workshop

1. **Comprendre** ce qu'est la **délégation Kerberos**, ses différents types et l'extension **S4U**.
2. **Exploiter** ces techniques.
   - Depuis Linux : NetExec, Impacket, BloodyAD ...
3. **Enchaîner** les attaques pour **compromettre** un domain Active Directory.
   - Énumération, Exploitation, Latéralisation.

Parti pris des workshops :

Volontairement non exhaustif, l'idée c'est de maîtriser la base.

ZLEB

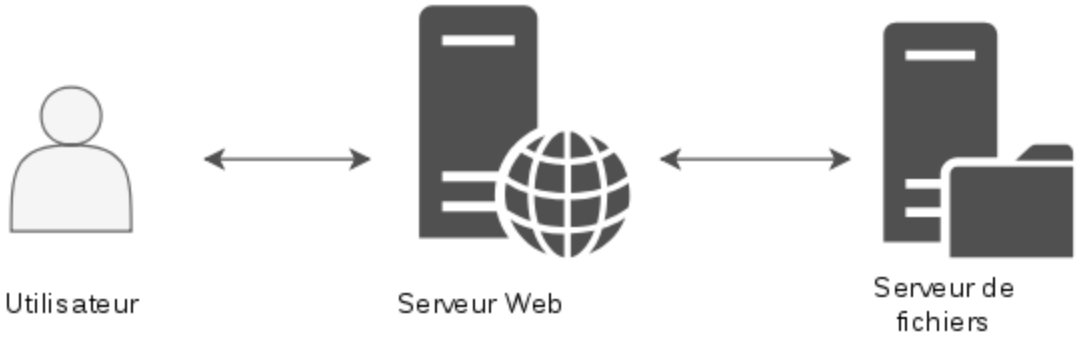
- ENSIBS 4A
- Apprentissage dans un bureau d'audit/pentest
- Certifié "Red Team Operator" - Zero Point Security

CQBAN

- ENSIBS 4A
- Apprentissage dans un SOC en orchestration agentique
- Certifié "Red Team Operator" - Zero Point Security

La Délégation ?

Cas usage :

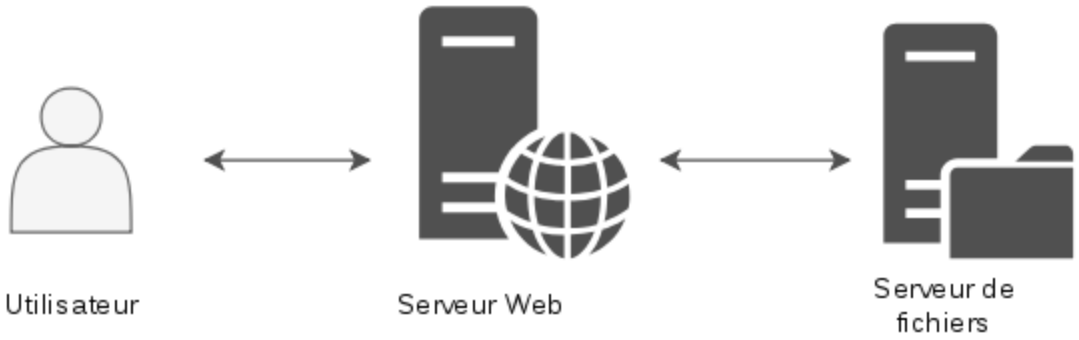


- **Utilisateur** : Se connecte au serveur web
- **Serveur WEB** : Serveur web affiche les fichiers d'un user
- **Serveur FICHIER** : Met à disposition les fichiers de l'utilisateur

Problème

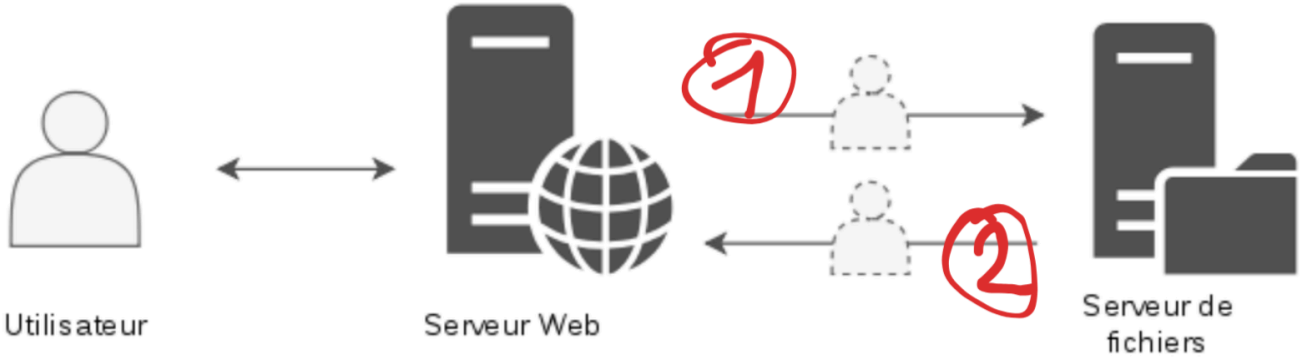
Le serveur web doit lire les perms du user, pour afficher les fichiers autorisées.

Problèmes ?



- C'est pas au serveur web de connaitre les perms du user sur le serveur de fichier.
- Le serveur web va **déléguer** l'authentification auprès du serveur de fichier.

Fonctionnement :



1. Le serveur web se connecte au serveur de fichiers comme s'il était l'utilisateur.
2. Le serveur renvoie uniquement les informations autorisées, puis le serveur web les affiche. Du point de vue du serveur de fichiers, c'est l'utilisateur qui fait la demande.

Trois types de délégation:

- Unconstrained Delegation : Le service peut se faire passer pour **n'importe quel utilisateur** auprès de **n'importe quel service**.
- Constrained Delegation : Le service peut déléguer uniquement **vers une liste de SPN définie**.
- Resource-Based Constrained Delegation : **La ressource cible décide** quels services peuvent déléguer vers elle.

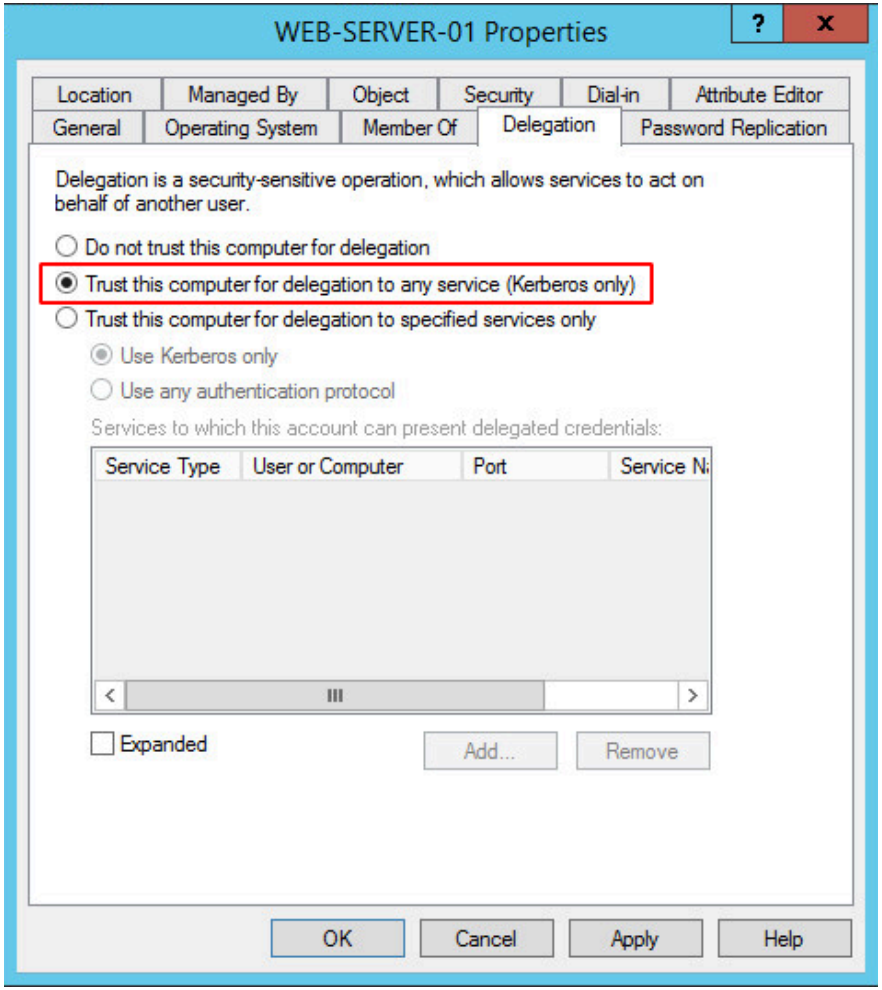
> La possibilité de relayer des identifiants peut être donnée à un compte possédant un SPN, donc un compte machine ou de service.

Unconstrained Delegation (KUD)

Unconstrained Delegation (KUD)

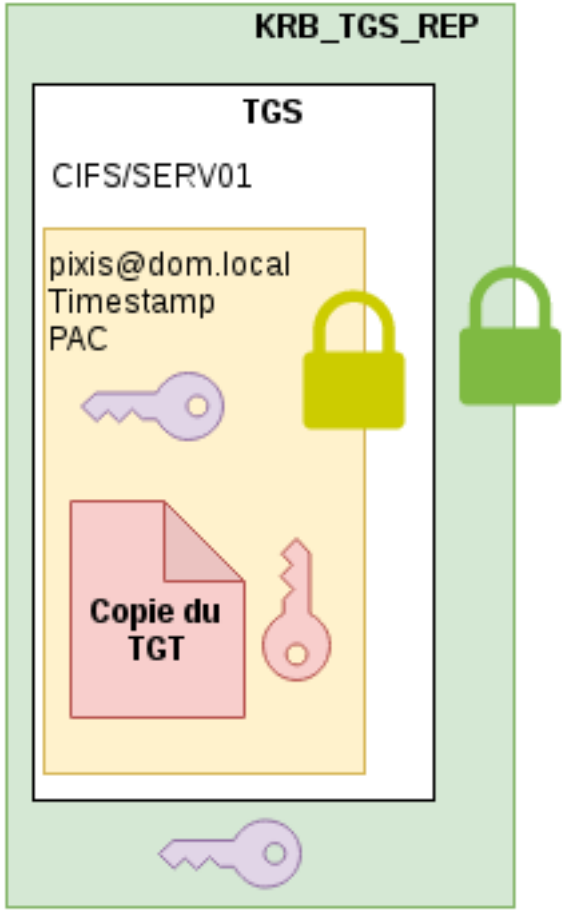
Le service peut se faire passer pour **n'importe quel utilisateur** auprès de **n'importe quel service** sur **n'importe quelle machine**.

- Flag UAC (User Account Control) `TRUSTED_FOR_DELEGATION` sur le serveur
- Flag `NOT_DELEGATED` **non positionné** sur l'utilisateur

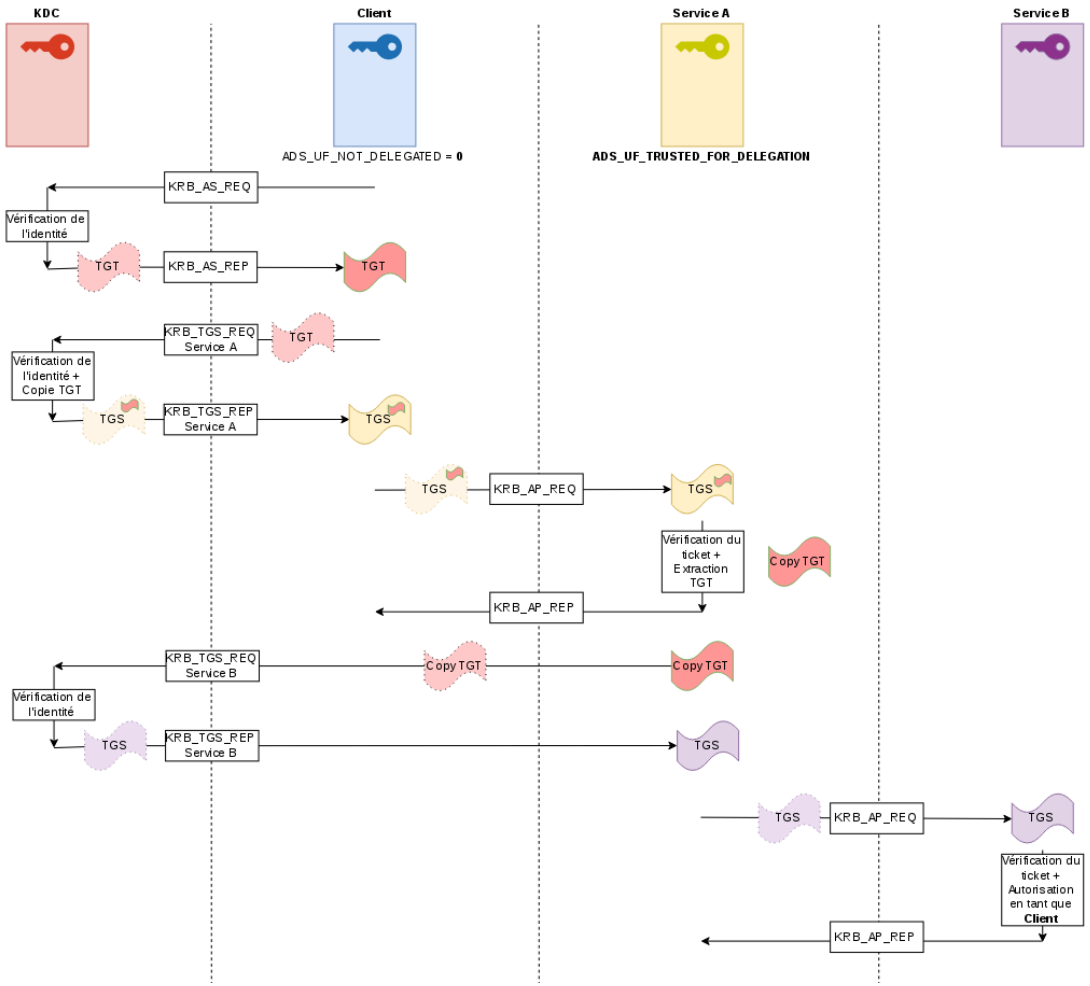


Fonctionnement:

KRB_TGS_REQ: Si les deux prérequis sont ok -> Le DC répond avec KRB_TGS_REP classique + copie du TGT de l'utilisateur et clé de session associée.



Unconstrained Delegation (KUD)



Extension de protocole

Objectif

Autorise un service à requêter un ticket au nom d'un autre utilisateur au KDC

Deux sous-protocols:

- S4U2Self (Service for User to Self)

- S4U2Proxy (Service for User to Proxy)

S4U2Proxy

Permet à un service d'obtenir un service ticket au nom d'un autre utilisateur.

S4U2Self

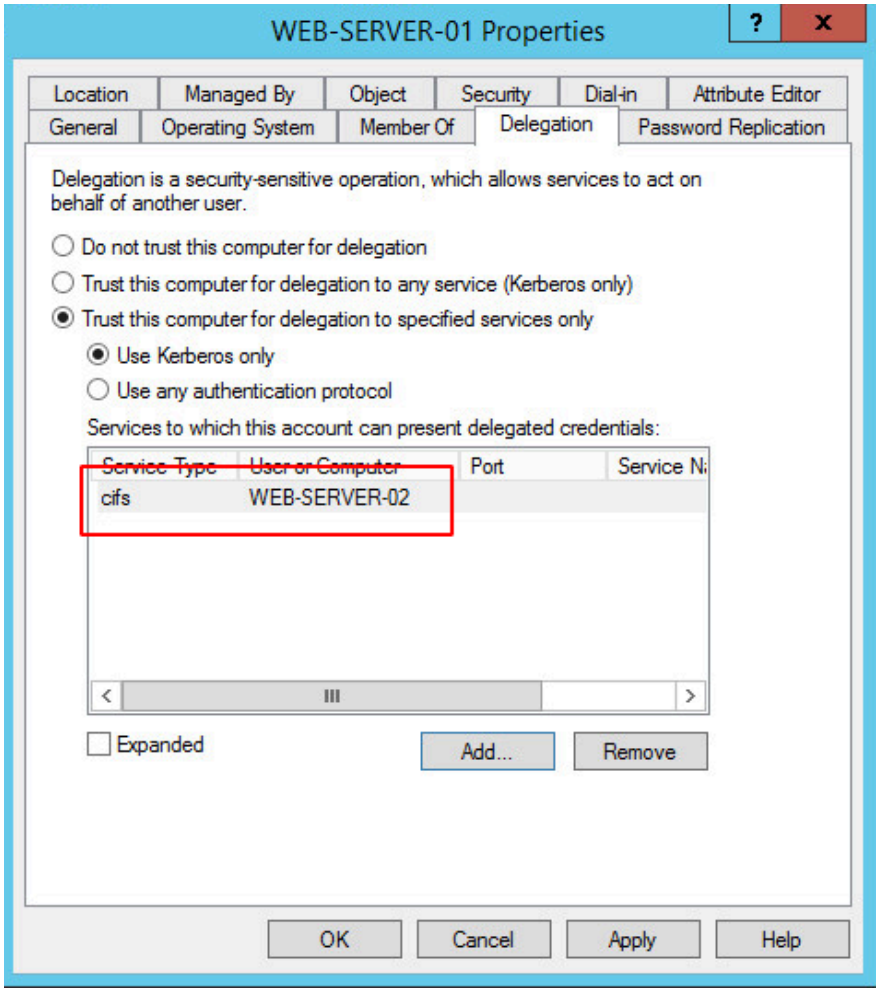
Permet à un service de demander un ticket au nom d'un utilisateur pour soit même.

Constrained Delegation (KCD)

Constrained Delegation (KCD)

Le service peut déléguer uniquement **vers une liste de SPN définie**.

- Attribut `msDS-AllowedToDelegateTo` avec liste de SPNs sur le serveur



Fonctionnement:

`Avec notre exemple User -> Web Server -> File Server`

- L'utilisateur s'authentifie sur le serveur web
- Service A doit accéder à Ressource B en tant que l'utilisateur (S4U2Self si besoin)
- Service A demande un ST au DC via l'extension S4U2Proxy

Deux champs clés dans le KRB_TGS_REQ pour S4U2Proxy :

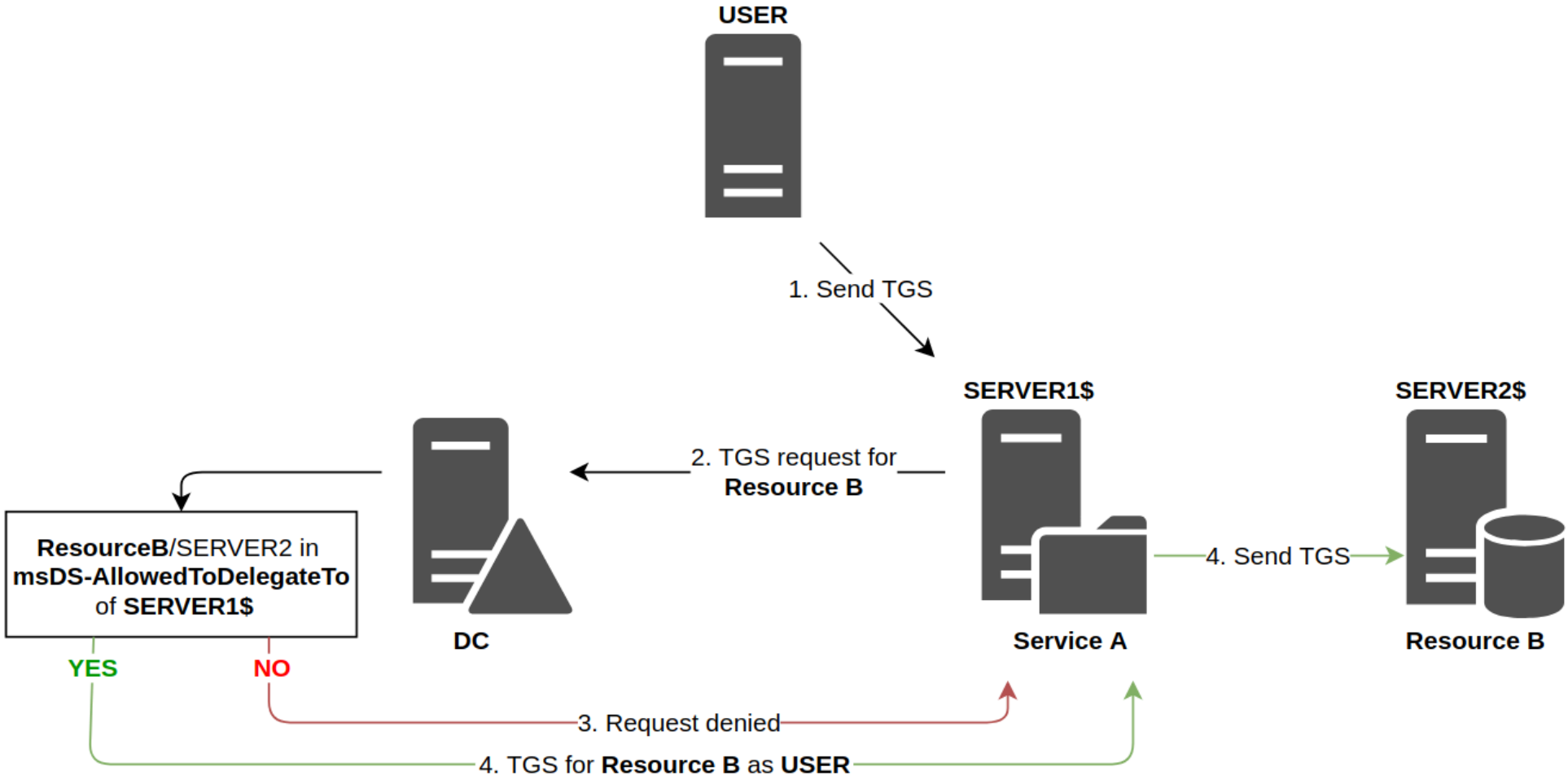
additional-tickets

Contient le ST de l'utilisateur (doit être forwardable ⟶ flag NOT_DELEGATED non positionné).

cname-in-addl-tkt

Flag,qui indique au DC d'utiliser l'identité du ticket dans additional-tickets plutôt que celle du serveur.

Constrained Delegation (KCD)

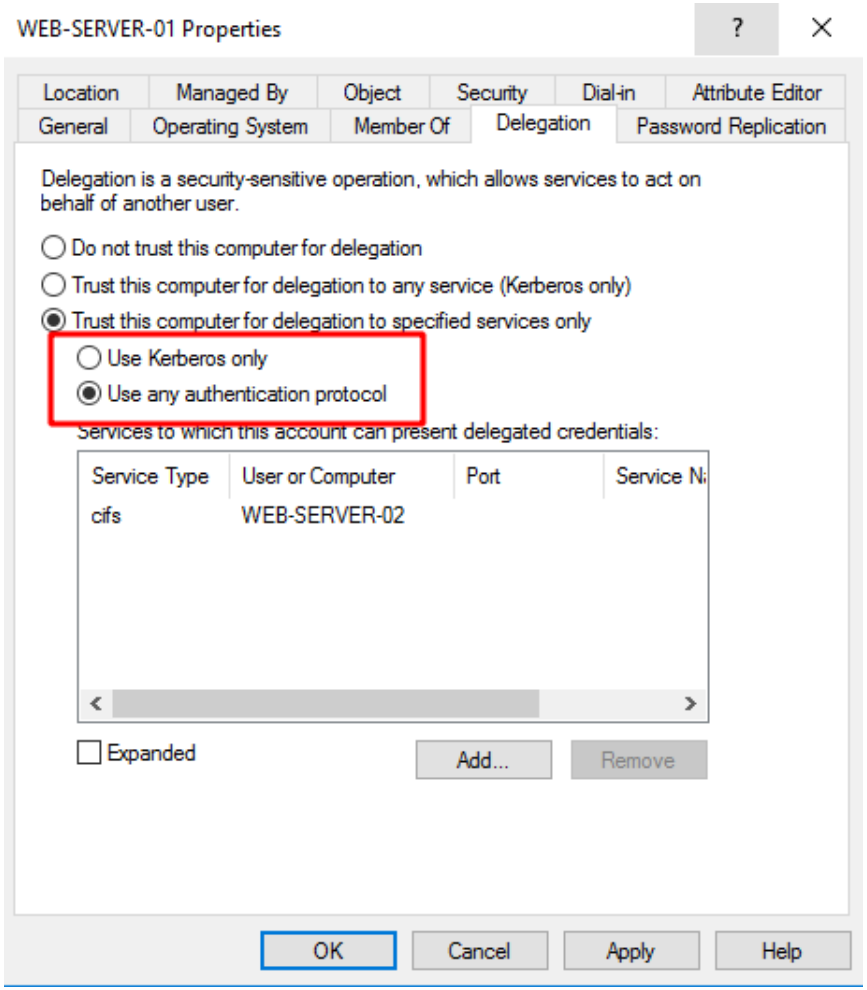


Protocol transition (S4U2Self)

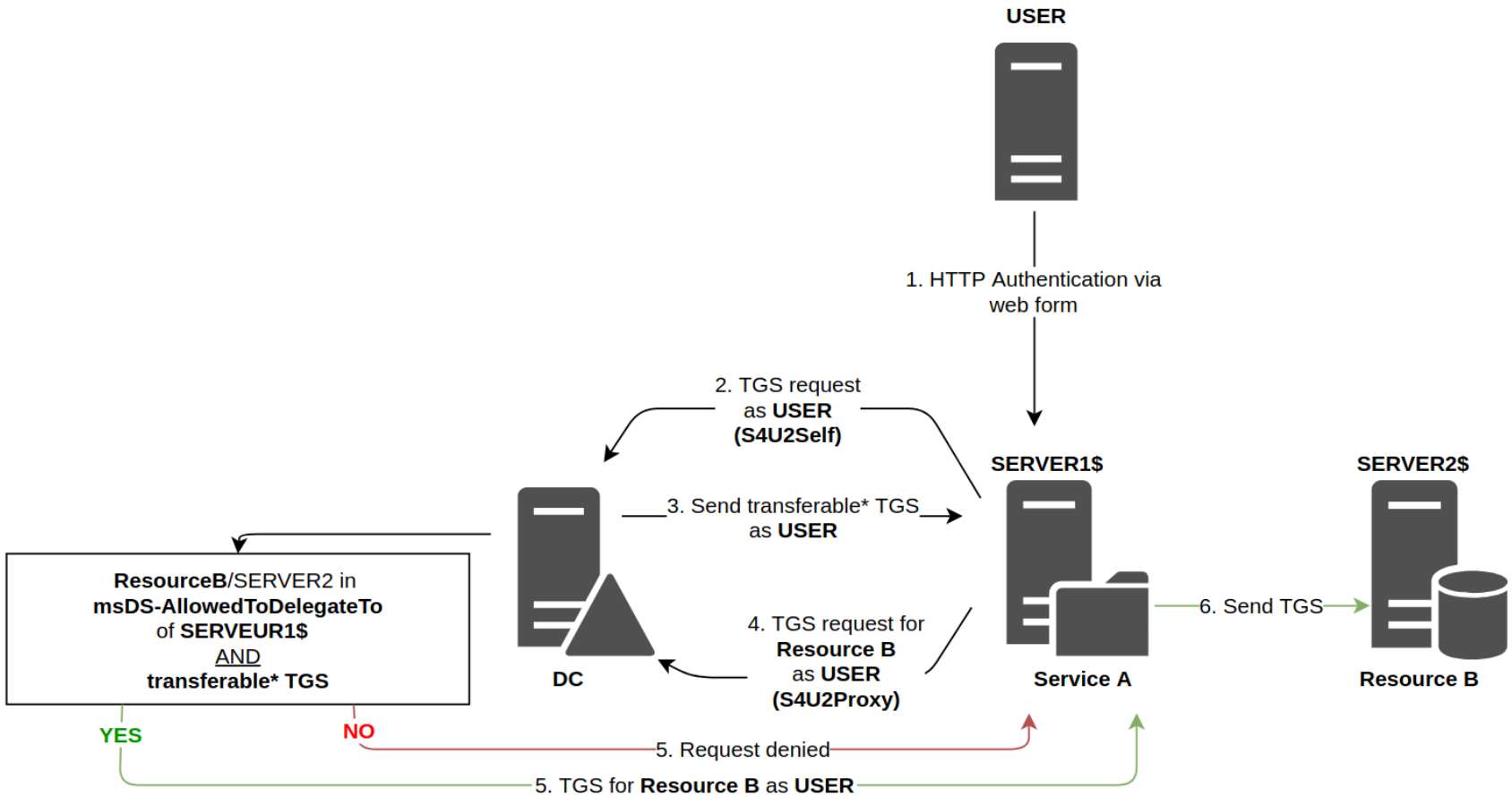
Le service peut obtenir un ticket Kerberos pour un utilisateur même si l'utilisateur ne s'est pas authentifié en Kerberos au départ.

Pourquoi c'est important ?

- Un User peut s'authentifier sur Service A **sans Kerberos** (NTLM, formulaire web...) → **Service A n'a pas de ST du User** à mettre dans `additional-tickets`
- **S4U2Self** : Service A **demande un ST pour son propre service au nom du user** avec le champ `PA-FOR-USER` dans le 1er KRB_TGS_REQ
- **S4U2Proxy** : Service A demande un ST pour le service backend → le ST obtenu via S4U2Self est placé dans additional-tickets du 2ème KRB_TGS_REQ



S4U2Self + S4U2Proxy



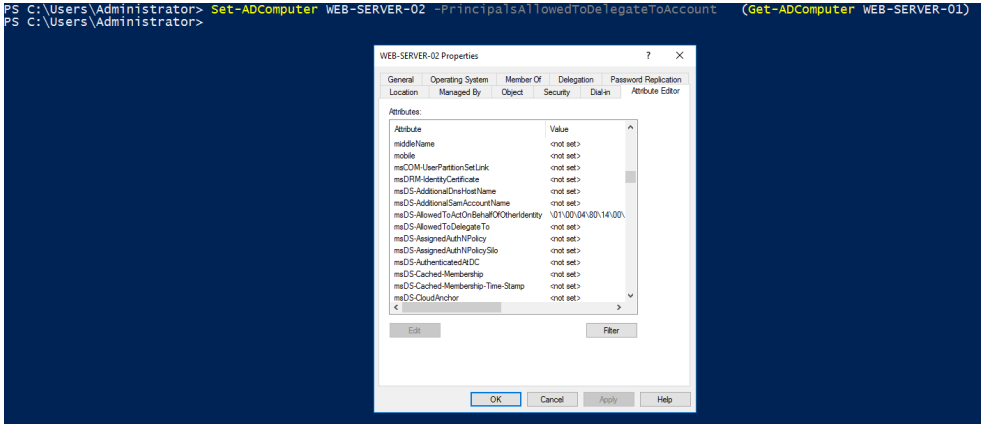
***transferable** : Only if TRUSTED_TO_AUTHENTICATE_FOR_DELEGATION flag is set

Ressource-Based Constrained Delegation (RBCD)

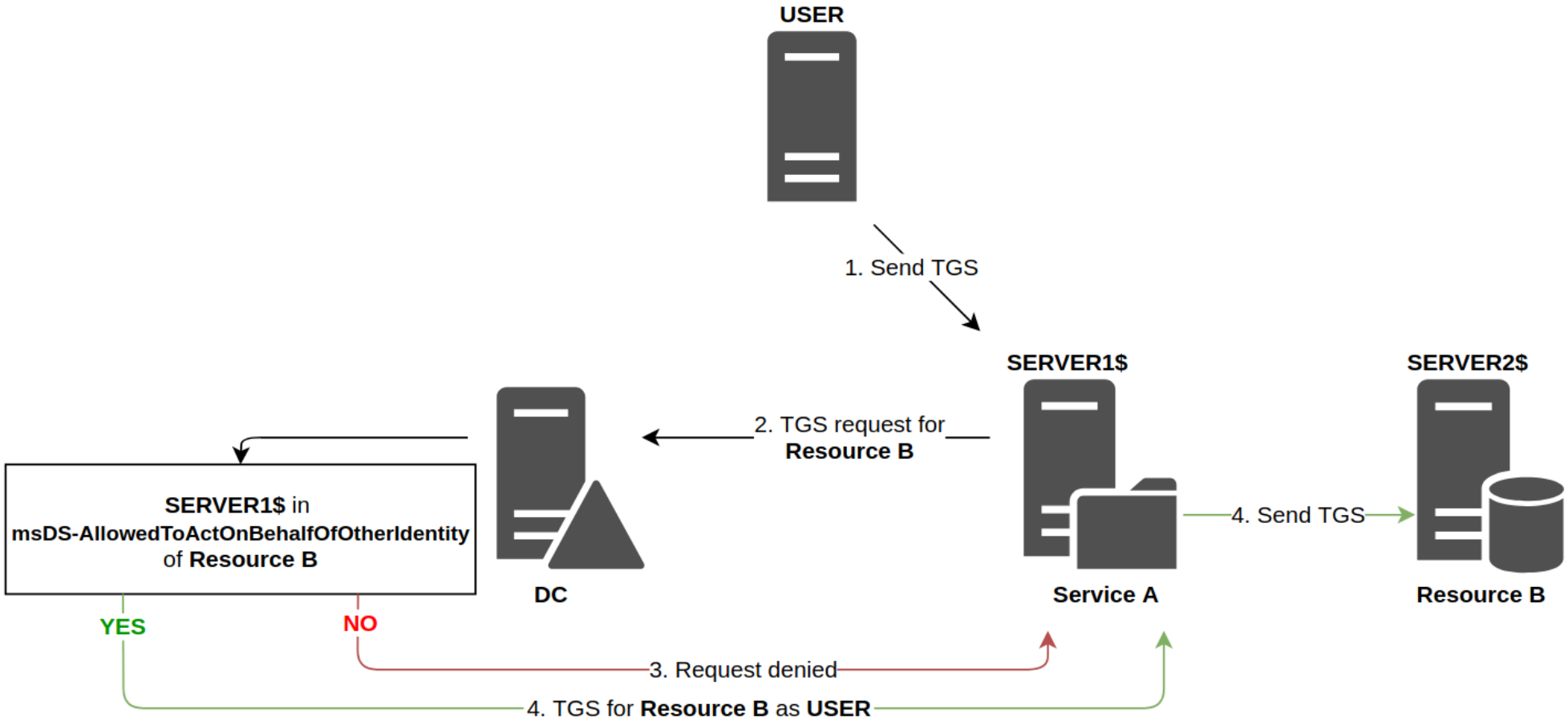
Ressource-Based Constrained Delegation (RBCD)

- **KCD** : le DC vérifie les attributs de Service A (*qui peut déléguer vers qui*)
- **RBCD** : le DC vérifie les attributs de Ressource B (*qui a le droit d'agir en son nom*)
- Attribut clé : `msDS-AllowedToActOnBehalfOfOtherIdentity` **sur le compte Machine Ressource B**
  - Contient la liste des comptes autorisés à s'y déléguer

  Ex : WEB-SERVER01 ajouté dans la liste de confiance de FILE-SERVER01



Ressource-Based Constrained Delegation (RBCD)



Ressources

Ressources

- https://beta.hackndo.com/constrained-unconstrained-delegation/
- https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-sfu/3bff5864-8135-400e-bdd9-33b552051d94g
- https://winprotocoldoc.z19.web.core.windows.net/MS-SFU/%5BMS-SFU%5D.pdf