Introduction au pentest Active Directory

Workshop GCC - Mai 2026

Zleb & Cqban

Objectifs du workshop

Objectifs du workshop

1. **Comprendre** des techniques d'attaques sur Active Directory.
   - AS-REP Roasting, Kerberoasting, DACL abuse.
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

Active Directory en bref

Définition :

- **Active Directory Domain Service (AD DS)**

> *Le système central qui gère les identités, les accès et les permissions dans un réseau Windows.*

- Extrêmement présent dans les SI d'entreprises

Composants :

Un annuaire (Directory)

Structure **hiérarchique** qui contient des informations sur des **objets**

- Forêt
- Domain
- Utilisateur
- Groupe
- Ordinateur

Composants :

Serveurs

Contrôleur de domaine : *le serveur central qui gère l'**authentification** et les **accès** de tous les objets du domaine.*

Objets

Toute entité **administrable** représentée dans l'annuaire : utilisateur, machine, groupe, ...

Protocoles - Authentification :

- Kerberos
  - Protocole “principal”, basé sur des tickets
- NTLM
  - Déprécié par Microsoft mais toujours utilisé en fallback

Dans les labs

On va s’intéresser aux attaques sur Kerberos, mais NTLM dans les labs.

Protocoles - Accès aux ressources :

- SMB
  - Accès aux ressources (fichiers) sur le réseau
- LDAP
  - Accès aux objet de l'annuaire

Dans les labs

Utile pour les étapes d'énumération. Souvent invisible c'est les tools qui les utilisent.

Kerberos

Objectif

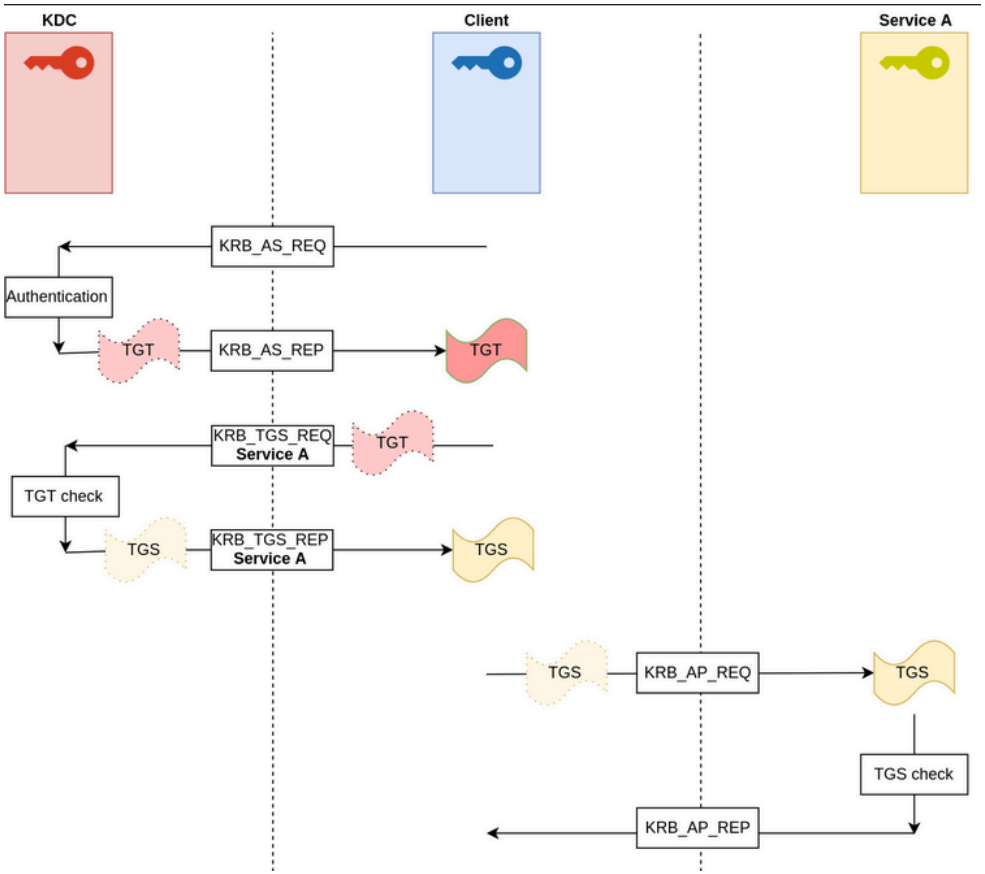
Permettre à des clients de s'authentifier sur le réseau afin d'accéder à des services sans communiquer leur mot de passe

Trois composants:

- Le client
- Le service
- un KDC (Key Distribution Center), un DC dans le cas d'AD

Accès au service en trois étapes:

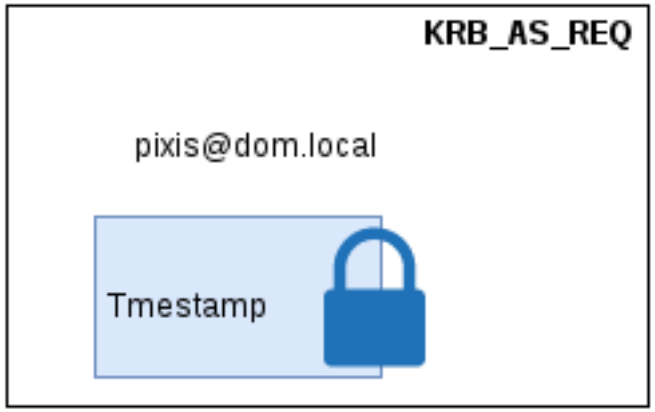
1. Authentification auprès du KDC -> TGT (Ticket Granting Ticket)
2. Demande de ticket pour le service -> ST (Service Ticket)
3. Accès au service grâce au ST



Contient:

- Username
- Challenge Timestamp

Préauthentification par timestamp chiffré avec une clé dérivée du hash NT de l'utilisateur si RC4, sinon de son mot de passe



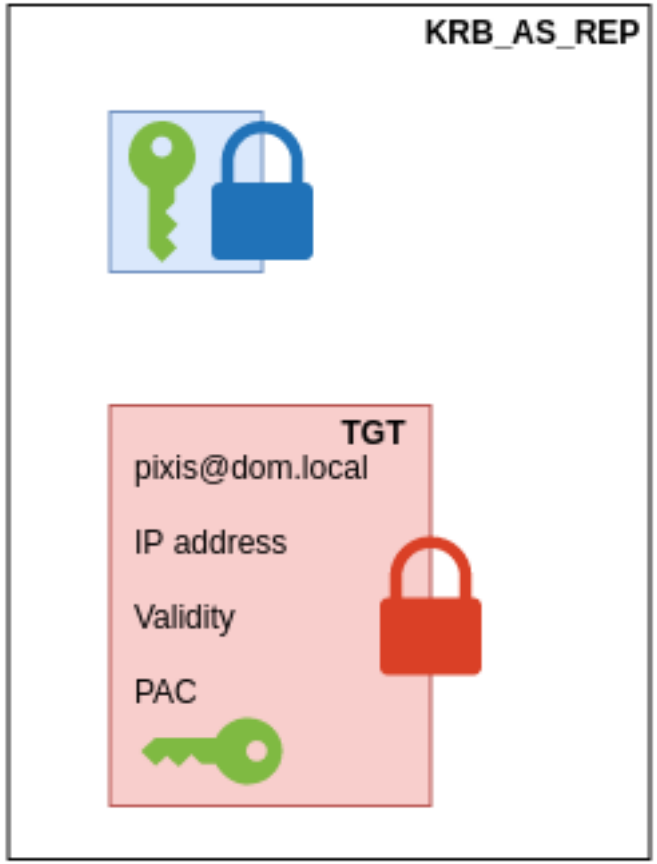
KRB_AS_REP

Si le client existe dans l'annuaire, le KDC tente de déchiffrer le timestamp, si la date correspond -> le KDC répond avec un KRB_AS_REP.

KRB_AS_REP

Contient:

- clé de session chiffrée avec le secret de l'utilisateur
- TGT chiffré avec le secret du KDC (compte krbtgt)
- Contient notamment :
  - Le username
  - La période de validité
  - La clé de session
  - Le Privilege Attribute Certificate (PAC) informations sur le client relatif à ses droits (ID, Groupes, ...)



La Pre-auth peut être désactivée sur un compte, dans ce cas, pas besoin du “challenge” avec le timestamp, n’importe qui peut alors demander un TGT au nom de cet utilisateur.

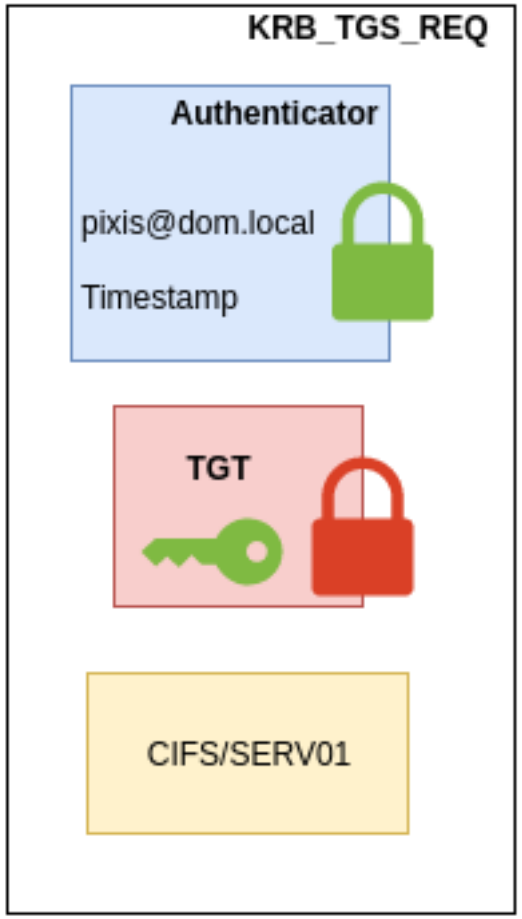
Mythe

Certaines vieilles applications ne fonctionneraient pas/mal avec la preauth, nécessitant de la désactiver.

KRB_TGS_REQ

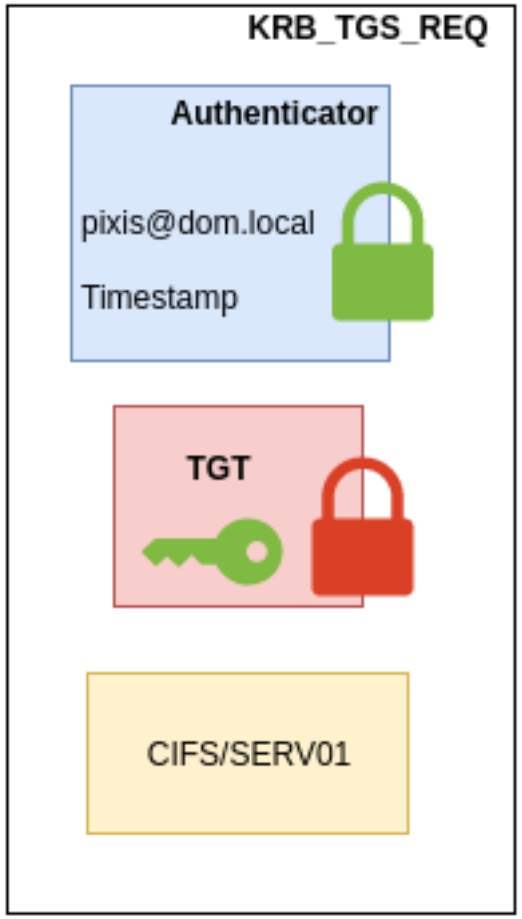
Demande de ST, contenant les informations suivantes:

- Un authentifiant contenant username et timestamp chiffrés avec la clé de session donnée par le KDC, permettant d'assurer que c'est bien le client qui fait la demande et que le TGT n'a pas juste été intercepté
- Le TGT
- Le SPN (Service Principal Name) du service voulu.



KRB_TGS_REQ

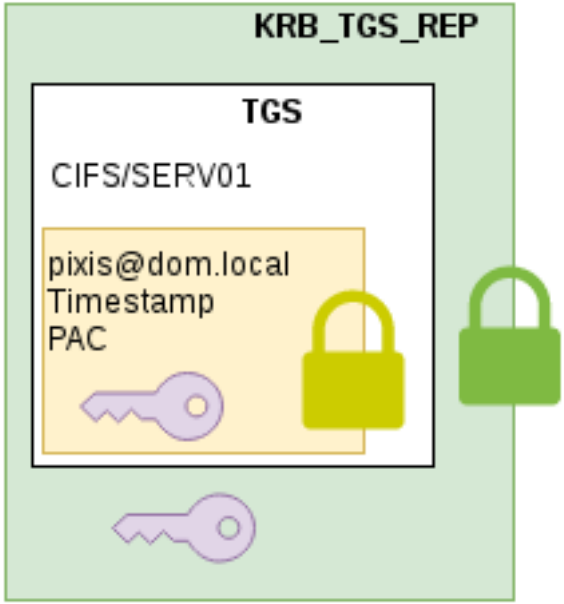
Une fois le KDC en possession de ces informations, il va alors déchiffrer le TGT, extraire la clé de session, déchiffrer l'authentifiant, et comparer son contenu avec celui du TGT afin de s'assurer de l'identité du client.



Si tout est ok, le KDC répond par un KRB_TGS_REP:

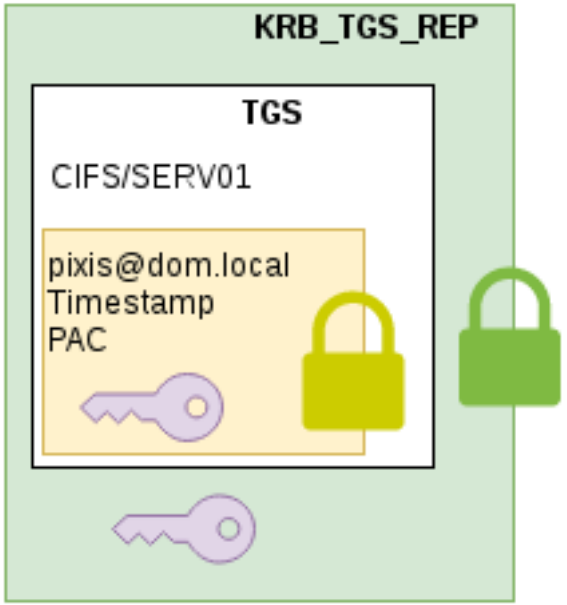
Contenant:

- Un Service Ticket (ST) contenant le SPN du service (`CIFS/SERV01`), le username, le PAC et une clé de session valide uniquement pour `pixis` discutant avec `CIFS` sur `//SERVER01` pendant un certain temps. Une **partie** du ticket est chiffrée avec le secret du compte de service.
- Et une copie accessible de la nouvelle clé de session pour communiquer avec le service



KRB_TGS_REP

Une fois le message reçu, le client peut alors le déchiffrer avec sa clé de session et récupérer la nouvelle clé ainsi que son ST.



Après un KRB_TGS_REP -> Client peut déchiffrer le message et en extraire un ST chiffré avec le secret du compte de service.

On peut alors tenter de bruteforce la partie chiffrée du ST afin de découvrir le mot de passe du compte de service.

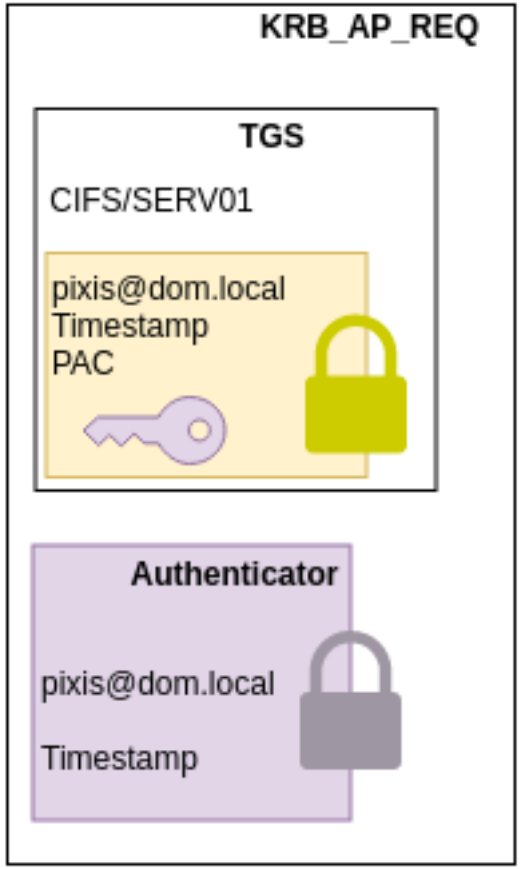
Vulnérabilité ?

Pas une vulnérabilité en soi. Le problème vient surtout de la faiblesse des mots de passe & le fait d'attribuer des SPN à des comptes utilisateurs qui ne disposent pas, par défaut, de mots de passe suffisamment robustes.

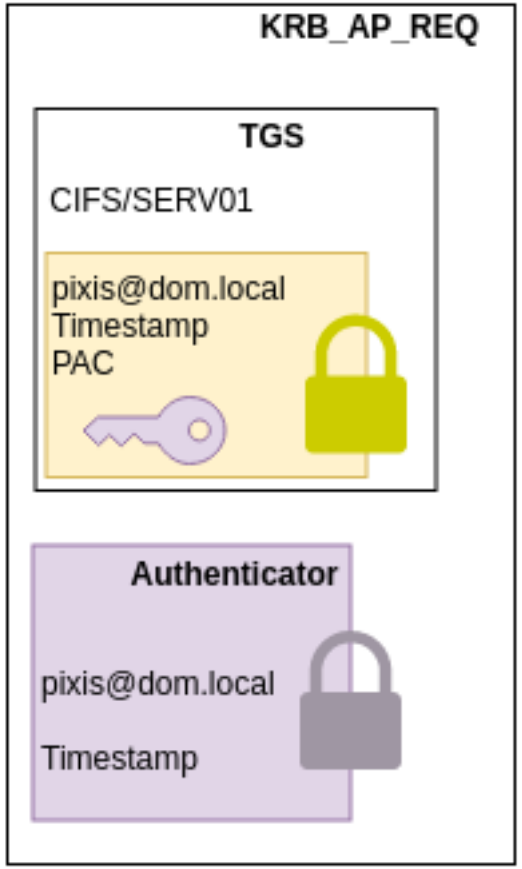
KRB_AP_REQ

Le client va ensuite envoyer un KRB_AP_REQ au service contenant les informations suivantes:

- Le ST
- Un nouvel authenticator chiffré cette fois-ci avec la clé de session utilisée entre le client et le service



le Service déchiffre le ST -> récupère clé de session -> déchiffre l'authentifiant -> compare les deux informations, si ok -> Le service accepte de communiquer avec le client et envoie un KRB_AP_REP.



DACL Abuse

Définition (encore) :

Security Descriptor

Structure qui contient les informations de sécurité d'un objet sécurisable.

- La grande majorité des objets dans l'AD sont sécurisables

C++ Copier

```
typedef struct _SECURITY_DESCRIPTOR {
  BYTE                        Revision;
  BYTE                        Sbz1;
  SECURITY_DESCRIPTOR_CONTROL Control;
  PSID                        Owner;
  PSID                        Group;
  PACL                        Sacl;
  PACL                        Dacl;
} SECURITY_DESCRIPTOR, *PISECURITY_DESCRIPTOR;
```

- SECURITY_DESCRIPTOR dans la Win32 API

Discretionary Access Control List (DACL)

Ensemble d'ACEs qui définit les *trustees* autorisés ou refusés à accéder à tout ou partie d'un objet.

Access Control Entry (ACE)

Règle spécifique d'accès sur un objet.

Anatomie d'une ACE

Trois dimensions :

- **Qui :** trustee/principal (identifiant utilisateur, groupe ..)
- **De quelle manière :** Action (lecture, écriture, accès complet) via l'*Access Mask*
- **Sur quoi :** l'objet ou un attribut

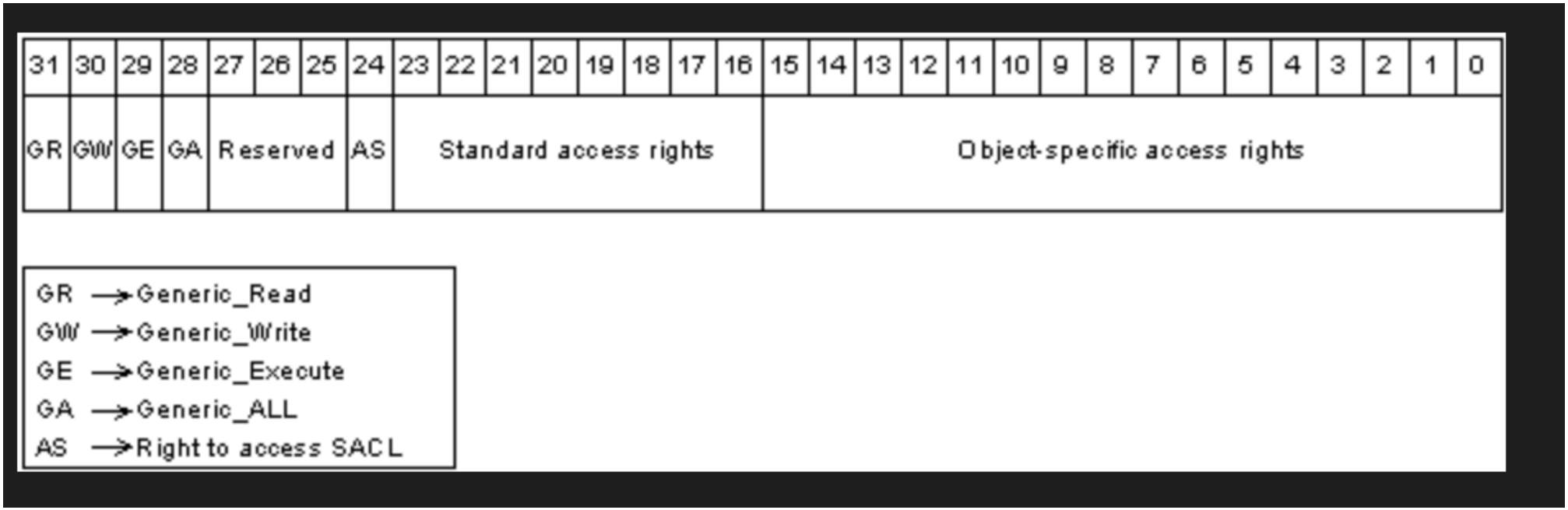
Dans les labs

Si un compte que l'on possède, détient tout ou parti d'un objet, grâce à une ACE : vecteur de latéralisation.

Type de droit d'accès

Access Mask

Valeur 32 bits qui encode les droits demandés ou accordés sur un objet dans une ACE.



Trois “catégories” :

- **Generic Access :** Lecture, Ecriture, Execution, TOUT -> **GenericAll**
- **Standard access :** Suppression, lire le SecDesc, Changer le propriétaire, ..
- **Object spécifique :** Lire un fichier, lire une clé de registre

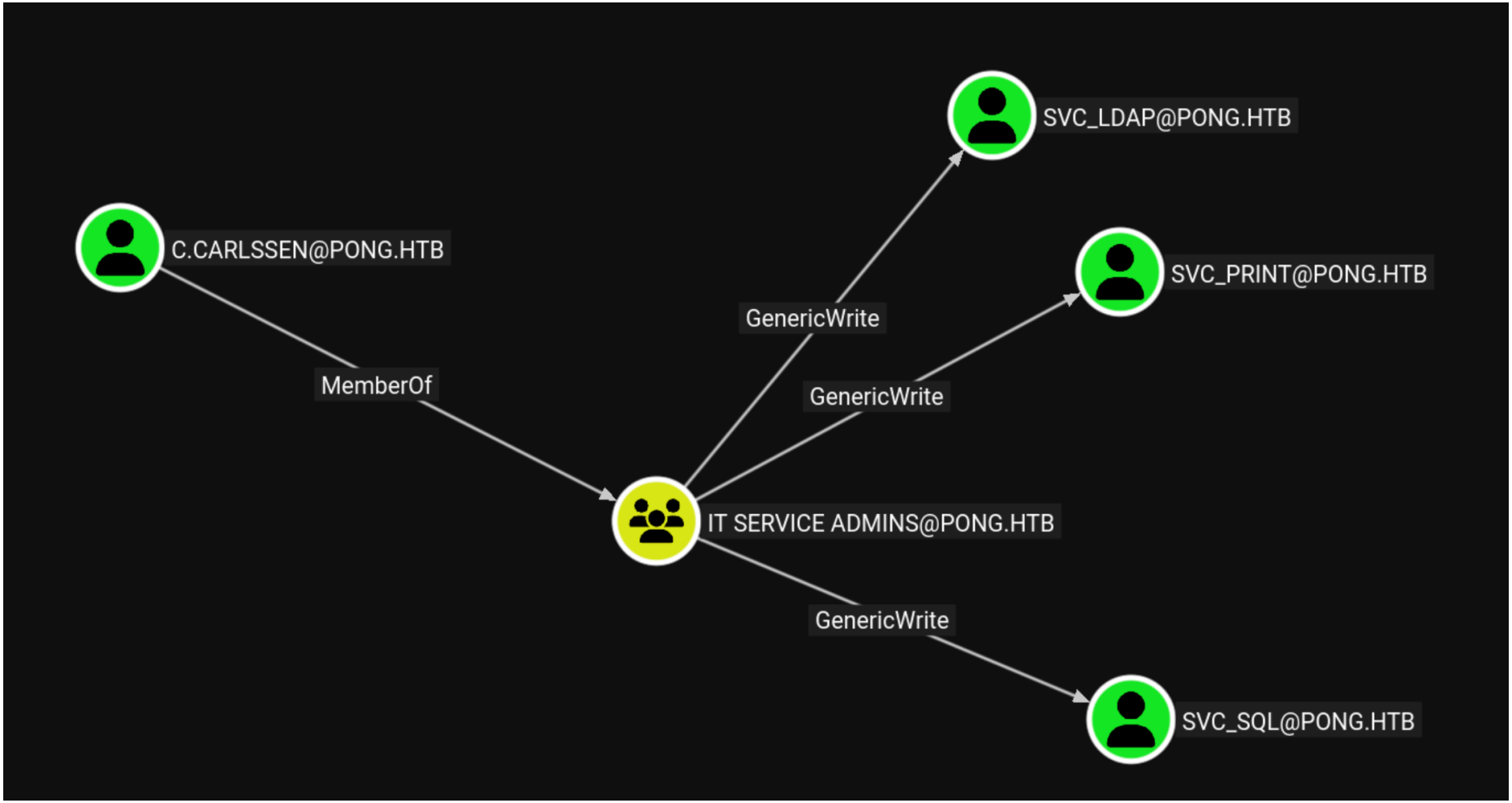
ForceChangePassword

This abuse can be carried out when controlling an object that has a `GenericAll`, `AllExtendedRights` or `User-Force-Change-Password` over the target user.

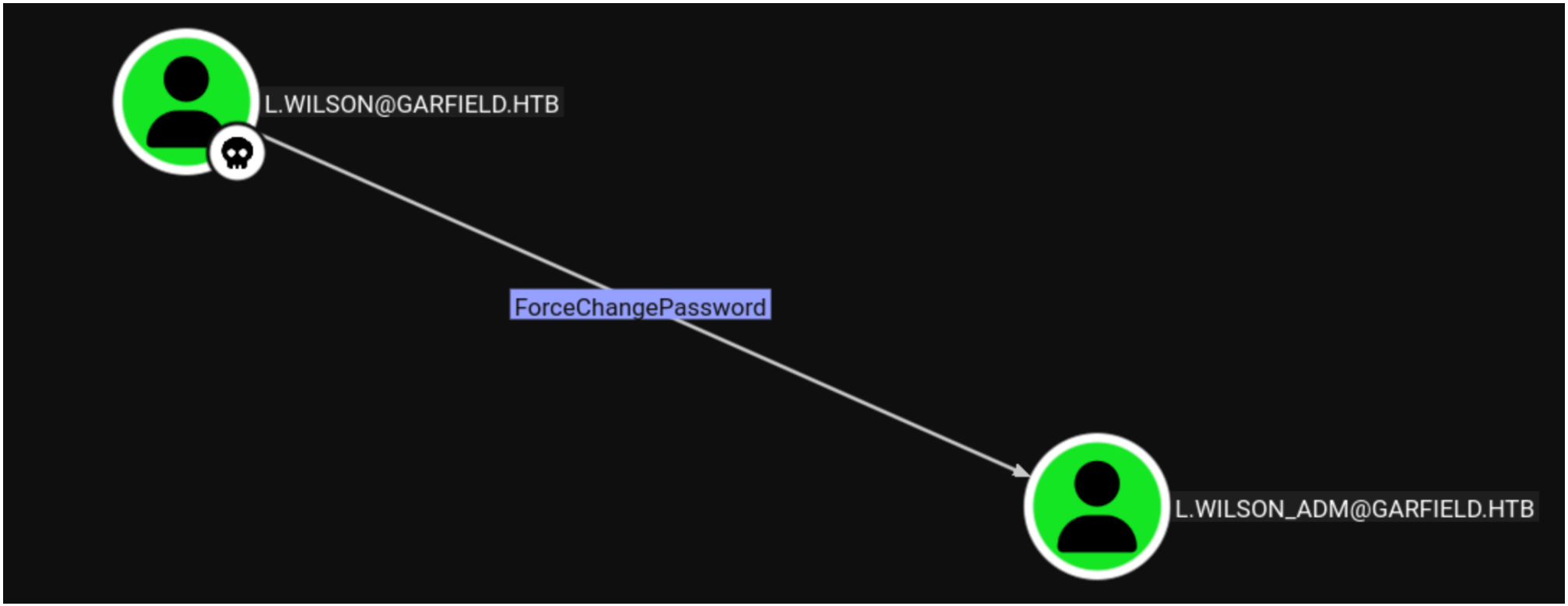
Targeted Kerberoasting

This abuse can be carried out when controlling an object that has a `GenericAll`, `GenericWrite`, `WriteProperty` or `Validated-SPN` over the target. A member of the Account Operator group usually has those permissions.

DACL Abuse


SVC_LDAP@PONG.HTB
C.CARLSSEN@PONG.HTB
SVC_PRINT@PONG.HTB
GenericWrite
MemberOf
GenericWrite
IT SERVICE ADMINS@PONG.HTB
GenericWrite
SVC_SQL@PONG.HTB

DACL Abuse


L.WILSON@GARFIELD.HTB
ForceChangePassword
L.WILSON_ADM@GARFIELD.HTB

Dans les labs

Enumeration via ldap, ingestion dans bloodhound, exploitation via impacket/ bloodyAD

Workshop 1 ...

Outils

- Distribution : exegol, kali linux
- Outils suivant : impacket,netexec, hashcat, bloodyAD, evil-winrm, SharpHound.exe, bloodhound CE

Ressources

Ressources

- Hackndo :
  - https://beta.hackndo.com/kerberos/
- The Hacker Recipes :
  - https://www.thehacker.recipes/ad/movement/dacl/
  - https://www.thehacker.recipes/ad/movement/kerberos/
- Documentation Microsoft :
  - https://learn.microsoft.com/en-us/windows/win32/secauthz/access-rights-and-access-masks
  - https://learn.microsoft.com/en-us/windows/win32/secauthz/dacls-and-aces
- Orange Cyber Défense Mind Map AD :
  - https://orange-cyberdefense.github.io/ocd-mindmaps/img/mindmap_ad_dark_classic_2025.03.excalidraw.svg