

Filière : Réseaux Informatiques Et Télécommunications

Option : Cybersécurité & Sécurité de l'information

Niveau : 4^{ième} Année

Rapport de projet en protocoles de sécurité

Projet d'authentification avec Kerberos



Réalisé par :

- Ayadhi Ibrahim
- Miled Ghassen
- Aloui Hichem

Professeur :

Youssfi Souheib

Table des matières

I-	Introduction générale	3
II-	Fonctionnement du protocole Kerberos.....	3
1.	Kerberos en Active Directory	3
2.	Fonctionnement.....	3
3.	Services SMB et CIFS	4
III-	Réalisation.....	5
1.	Configuration du Windows Server 2016	5
1.1-	Configuration du nom de la machine et de l'adresse IP :	5
1.2-	Installation du serveur DNS et des rôles Active Directory :	6
1.3-	Configuration AD DS.....	7
1.4-	Configuration DNS.....	8
1.5-	Installation et configuration du serveur NTP	9
2.	Configuration de la machine Serveur.....	11
2.1-	Configuration d'une IP statique et du nom de la machine :	11
2.2-	Configuration du DNS.....	11
2.3-	Configuration NTP	12
2.4-	Configuration SMB/Winbind	12
2.4.1-	Installation des paquets nécessaires	12
2.4.2-	Fichiers de configuration.....	12
2.5-	Configuration des dossiers partagés	14
3.	Configuration de la machine Client.....	15
4.	Authentification Kerberos	15
4.1-	Joindre le royaume :	15
4.2-	Obtention TGT.....	16
4.2.1-	Partie théorique :	16
4.2.2-	Partie pratique :	17
4.3-	Obtention TGS et accès au service	18
4.3.1-	Partie théorique	18
4.3.2-	Partie pratique :	22
4.4-	Résumé.....	24
IV-	Conclusion générale.....	25

I- Introduction générale

Kerberos est un protocole d'authentification AAA issu du projet « Athena » du MIT (Massachusetts Institute of Technology). Il est chargé d'authentifier, d'autoriser et de surveiller les utilisateurs voulant accéder aux ressources et services de votre réseau.

- L'authentification désigne le fait de prouver qu'on est bien la personne que l'on prétend être. L'authentification vient en complément de l'identification. Pour s'authentifier, on ajoute une preuve à l'identification.
- L'autorisation est la deuxième phase de la triade AAA. Elle agit une fois que l'utilisateur s'est authentifié. C'est dans cette phase qu'on donne ou non accès à la ressource demandée, en fonction de la politique de contrôle d'accès.
- La dernière des trois phases de la triade AAA est désignée par le terme Accounting qui peut être traduit par traçabilité dans ce contexte. Les utilisateurs se sont authentifiés, puis ont obtenu une autorisation d'accès. Maintenant on garde une trace de toutes les actions effectuées par l'utilisateur. On dit que les actions de l'utilisateur sont loguées. Un administrateur réseaux pourra ainsi, consulter les logs afin de vérifier les actions d'un utilisateur, ou bien retrouver l'auteur de telle ou telle action.

Le protocole **Kerberos** a été normalisé dans sa version 5 par l'IETF dans les RFC 1510. C'est un standard qui résout de nombreux problèmes de sécurité, d'administration, et de productivité dans l'authentification des clients et des services au sein d'un réseau. En effet, **Kerberos** introduit le principe de Single Sign-On (SSO). Ainsi avec une authentification unique, l'utilisateur aura accès à tous les services du réseau.

Le **SSO**, pour Single Sign-On, désigne un système d'authentification permettant à un utilisateur d'accéder à de nombreuses applications sans avoir à multiplier les authentifications. L'utilisateur renseigne un mot de passe en début de session et peut ensuite accéder à de nombreuses applications informatiques sans être contraint de devoir s'authentifier sur chacune d'entre elles

II- Fonctionnement du protocole Kerberos

1. Kerberos en Active Directory

Active Directory est une solution de Microsoft utilisée pour la gestion d'un système d'information, articulée sur les points suivants :

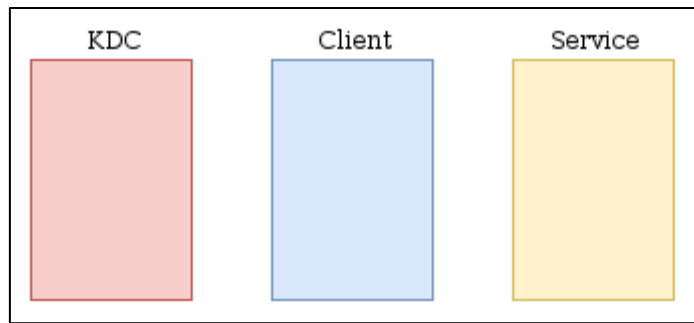
- Un système d'annuaire de ressources (LDAP)
- Un système d'authentification (Kerberos)
- Un système de résolution de noms (DNS)
- Une politique logicielle homogène

Nous allons nous intéresser dans cet article à la partie authentification au sein d'un Active Directory, donc à la partie Kerberos.

2. Fonctionnement

Le besoin auquel répond Kerberos est celui d'un utilisateur qui souhaite utiliser un service exposé quelque part sur le réseau, sans pour autant que l'utilisateur ait besoin d'envoyer son mot de passe, et sans que le serveur ait besoin de connaître les mots de passe de tout le monde. C'est une authentification centralisée. Pour répondre à cette problématique, il faut au minimum trois entités

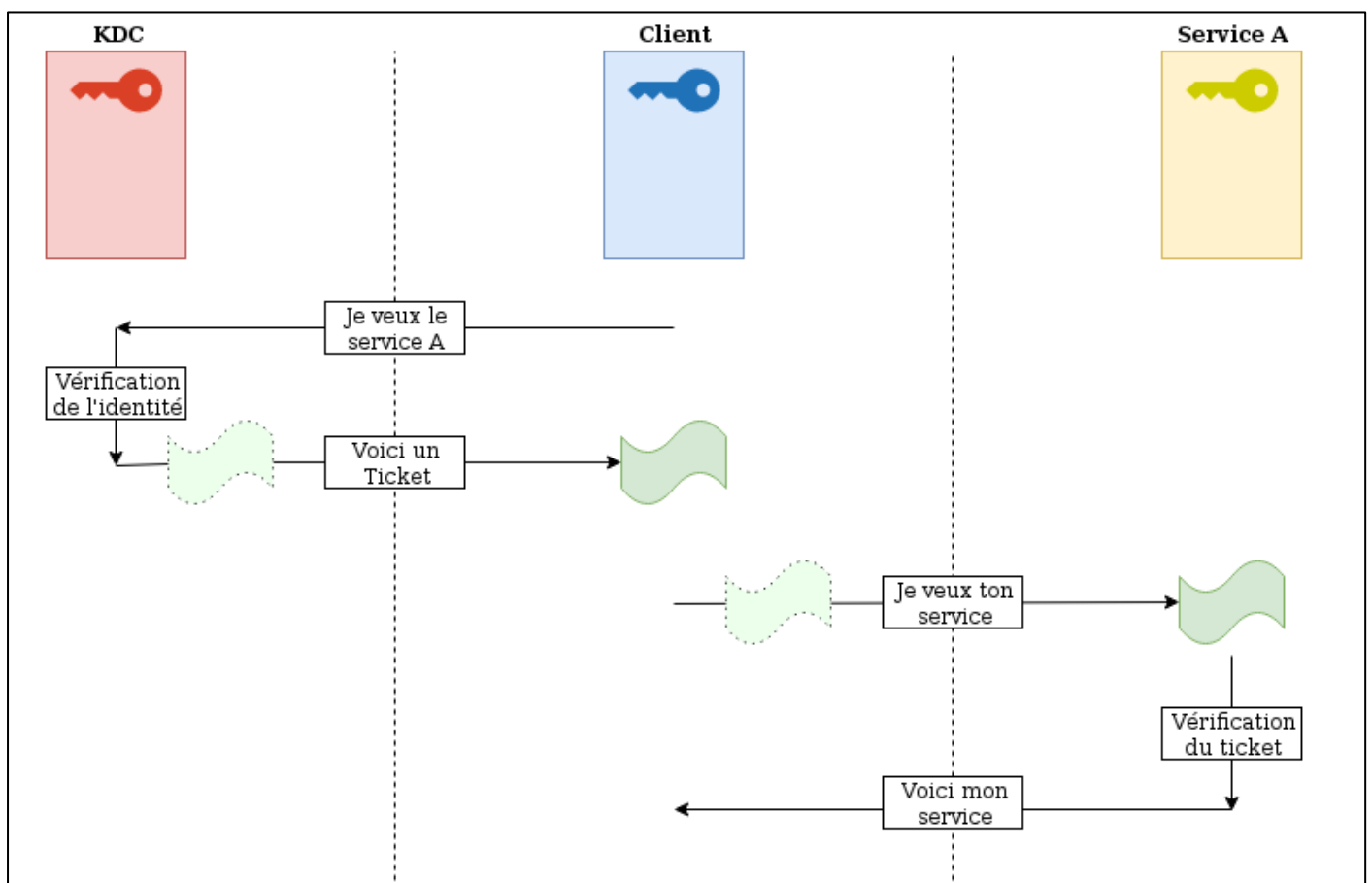
- Un client, qui peut être un ordinateur, un service, une personne, ...
- Une machine proposant un service
- Un *Key Distribution Center* ou centre de distribution de clés (KDC) qui est le contrôleur de domaine (DC) en environnement Active Directory



L'idée est que lorsque le client veut accéder au service, aucun mot de passe ne sera envoyé sur le réseau, évitant ainsi des fuites de ceux-ci qui pourraient compromettre le réseau, et l'authentification est centralisée, c'est au niveau du KDC que ça se passe.

Pour cela, le processus est un peu lourd, et se découpe en trois étapes :

1. **Authentication Service (AS)** : Le client doit s'authentifier auprès du KDC
2. **Ticket-Granting Ticket (TGT)** : Il doit ensuite demander un ticket permettant d'accéder au service choisi (Dans notre cas, c'est : CIFS)
3. **Accès au service (AP)** : Il communique enfin avec le service en lui fournissant le ticket



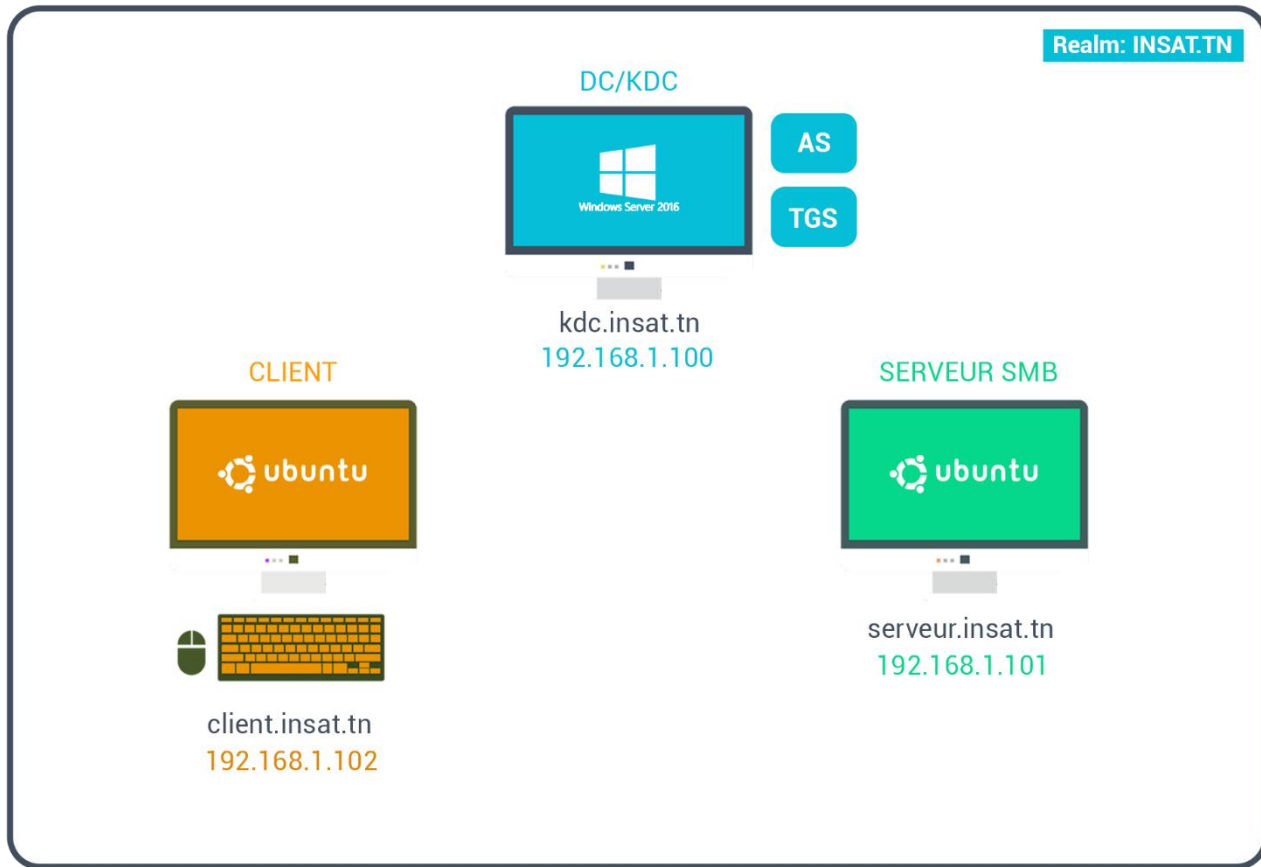
3. Services SMB et CIFS

- **SMB** : Le sigle SMB signifie “*Server Message Block*”. Il s’agit d’un protocole de partage de fichiers inventé par IBM. Il a été conçu pour permettre aux ordinateurs de lire et d’écrire des fichiers sur un hôte distant via un système de réseau local (LAN).
- **CIFS** : CIFS est le sigle de “*Common Internet File System*”. **CIFS est un dialecte de SMB**. Plus clairement, *CIFS est une mise en œuvre particulière du protocole SMB, créée par Microsoft*.
- **Quelle est la différence ?** : Dans la plupart des cas, lorsqu’on dit qu’on utilise SMB ou CIFS, on parle de la même chose. Les deux versions du protocole sont équivalentes tant du point de vue intellectuel que du point de vue pratique : une machine client « parlant » le CIFS peut dialoguer avec un serveur parlant le SMB et vice versa

III- Réalisation

Notre projet se compose d'un KDC (Windows Server 2016), un client et un serveur (Ubuntu 18.04 LTS).

Voici un résumé de ce qu'on va implémenter



1. Configuration du Windows Server 2016

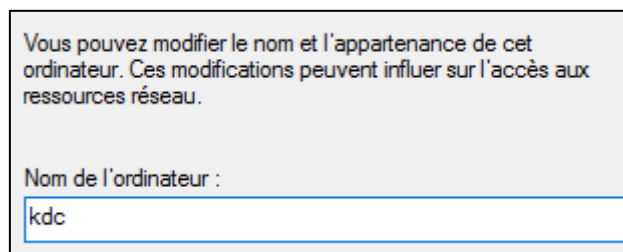
1.1- Configuration du nom de la machine et de l'adresse IP :

- **Modification du nom de la machine**

Ouvrir le « Gestionnaire de serveur » puis cliquer sur « Serveur Local »

Cliquer sur le nom de l'ordinateur (WIN-XXXXXXXXXX).

La fenêtre « Propriétés système » s'ouvre, cliquer sur « Modifier ».



Modifier le nom de votre ordinateur puis cliquer sur « OK ».

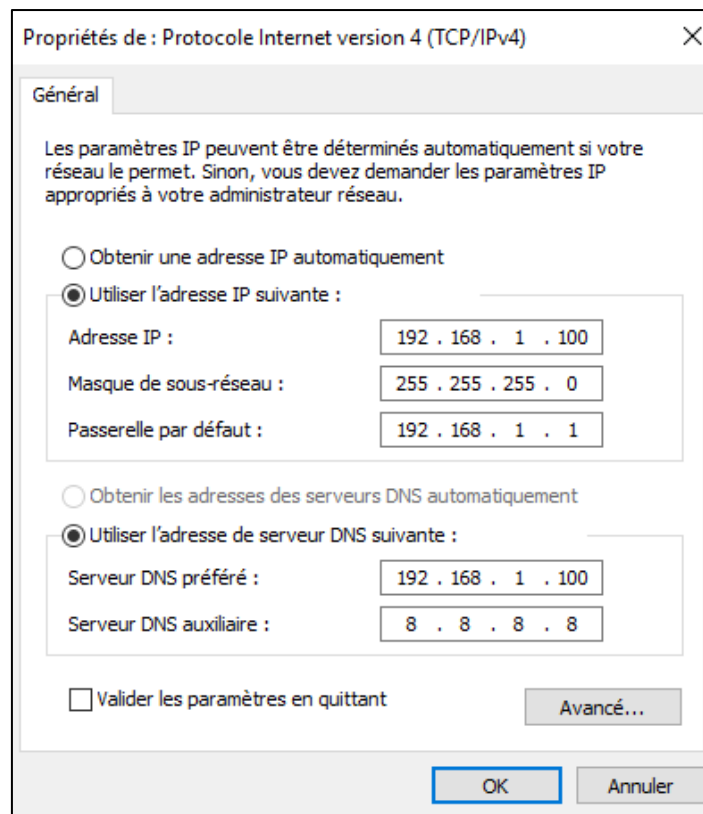
- **Définir une adresse IP fixe**

Ouvrir le « Gestionnaire de serveur » puis cliquer sur « Serveur Local ».

Cliquer sur « Adresse IPv4 attribuée par DHCP, Compatible IPv6 ».

Faire un clic droit sur la carte Ethernet0 puis cliquer sur « Propriétés ».

Sélectionner « Protocole Internet version 4 (TCP/IPv4) puis cliquer sur « Propriétés »



Nous allons maintenant redémarrer la machine.

1.2- Installation du serveur DNS et des rôles Active Directory :

Ouvrir le « Gestionnaire de serveur » puis cliquer sur « Ajouter des rôles et des fonctionnalités ».

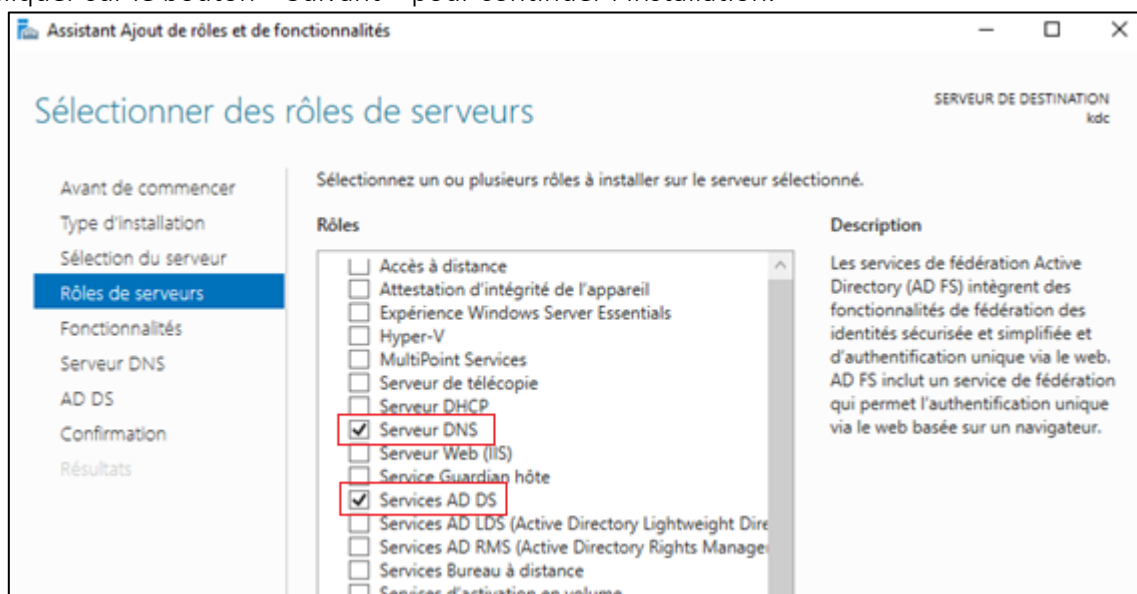
La fenêtre « Assistant Ajout de rôles et de fonctionnalités » s'ouvre. Cliquer sur le bouton « Suivant »

Cocher « Installation basée sur un rôle ou une fonctionnalité » puis cliquer sur « Suivant ».

Sélectionner votre serveur, puis cliquer sur « Suivant ».

Cocher les rôles « AD DS » et « DNS » et valider quand l'assistant vous propose d'installer les outils de gestion. Qui dit outils de gestion, dit console d'administration comme "Utilisateurs et ordinateurs Active Directory" mais aussi le module PowerShell pour Active Directory.

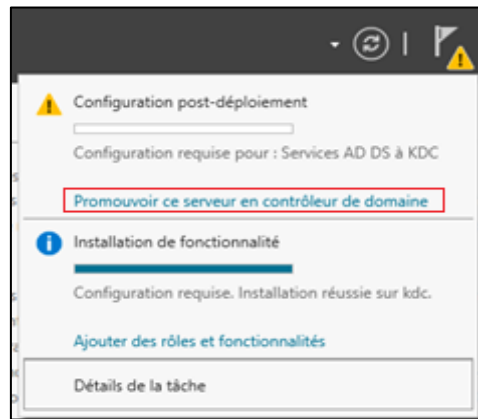
Ensuite, cliquer sur le bouton « Suivant » pour continuer l'installation.



Sur toutes les fenêtres suivantes, cliquer sur « Suivant », puis « Installer ».

Une fois l'installation terminée, sur la page « Gestionnaire de serveur », cliquer sur le drapeau avec un triangle jaune.

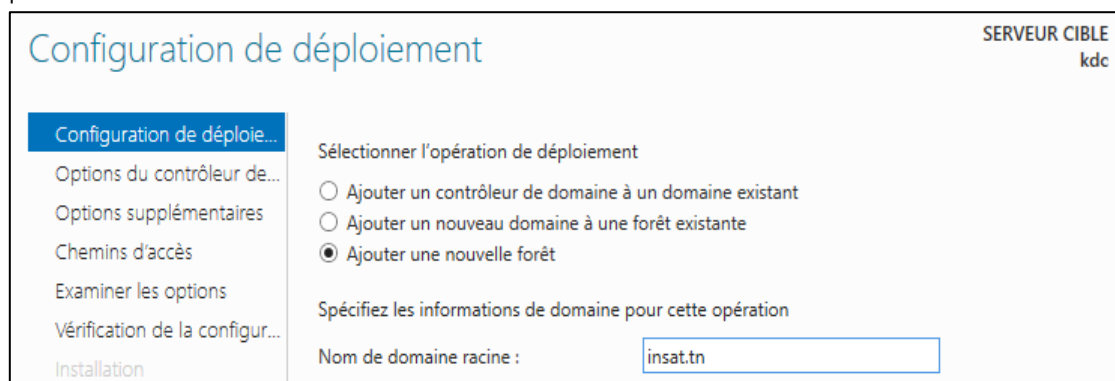
Cliquer ensuite sur « Promouvoir ce serveur en contrôleur de domaine » pour commencer la configuration de déploiement.



1.3- Configuration AD DS

Comme il s'agit d'un nouveau domaine dans une nouvelle forêt, Cocher « Ajouter une nouvelle forêt » et renseigner un nom dans le champ « Nom de domaine racine ».

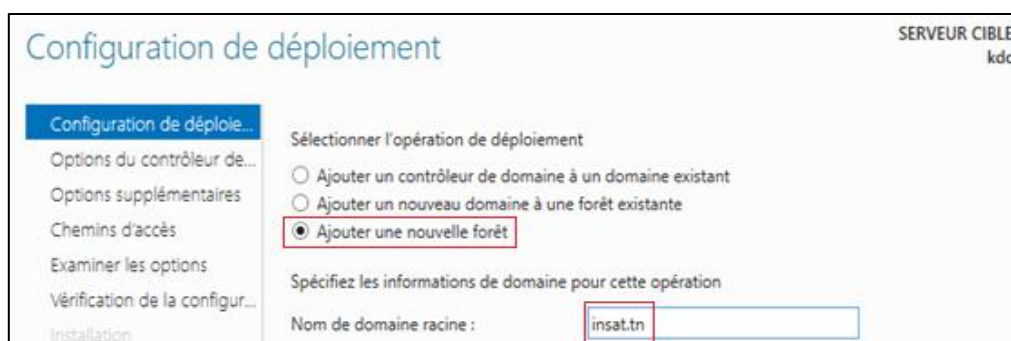
Ensuite cliquer sur le bouton « Suivant ».



Taper un mot de passe pour le mode de restauration des services de d'annuaire (DSRM), puis cliquer sur « Suivant ».

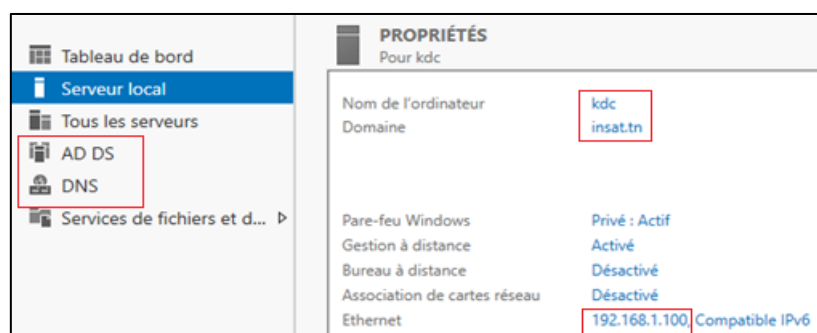
Sur la fenêtre de l'option DNS, cliquer sur « Suivant ».

Vérifier votre nom de domaine du NetBIOS puis cliquer sur « Suivant ».

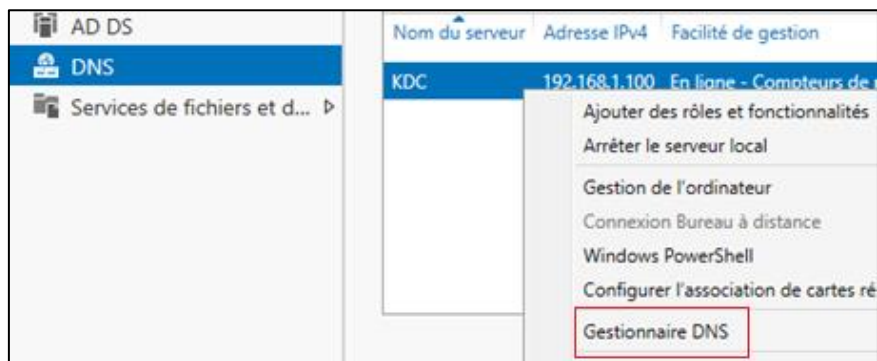


Sur toutes les fenêtres suivantes, cliquer sur « Suivant », puis « Installer ».

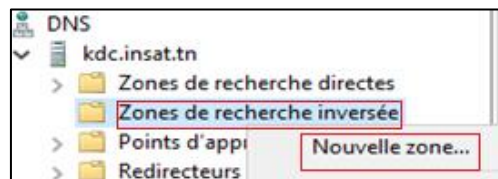
Une fois l'installation terminée, votre machine va redémarrer automatiquement.



1.4- Configuration DNS



- Créer zone de recherche inversée :



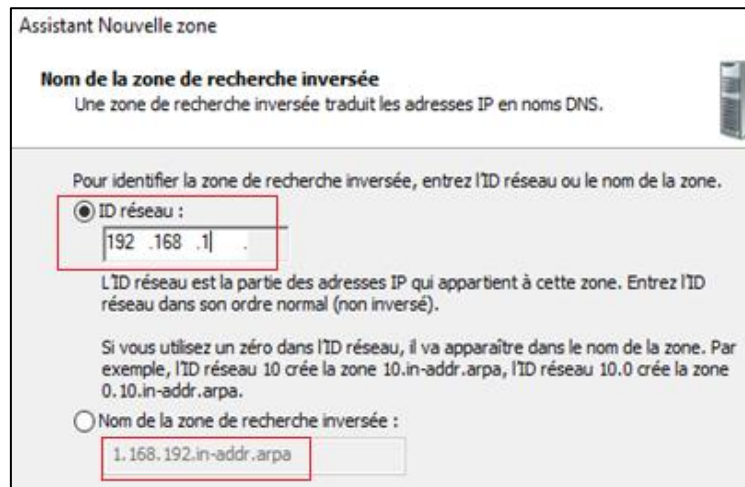
Choisir « vers tous les serveurs DNS exécutés sur des contrôleurs de domaine dans cette forêt »

Cliquer Suivant

Choisir « Zone principale »

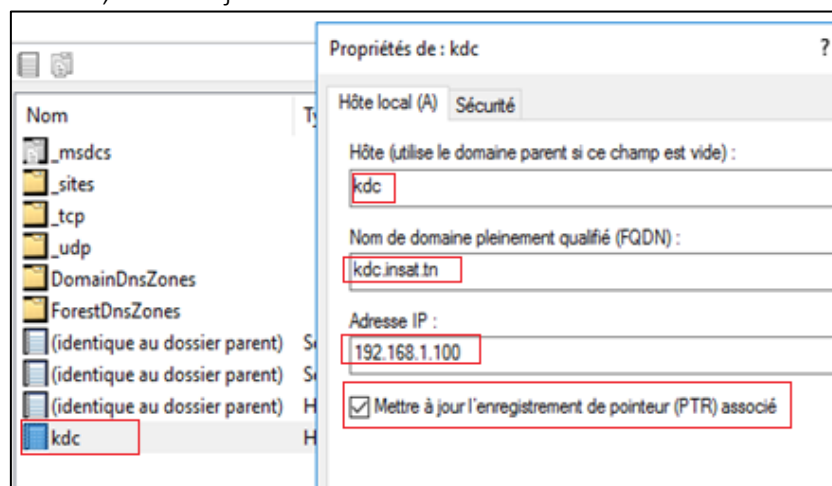
Cliquer Suivant

Mentionner votre sous-réseau :

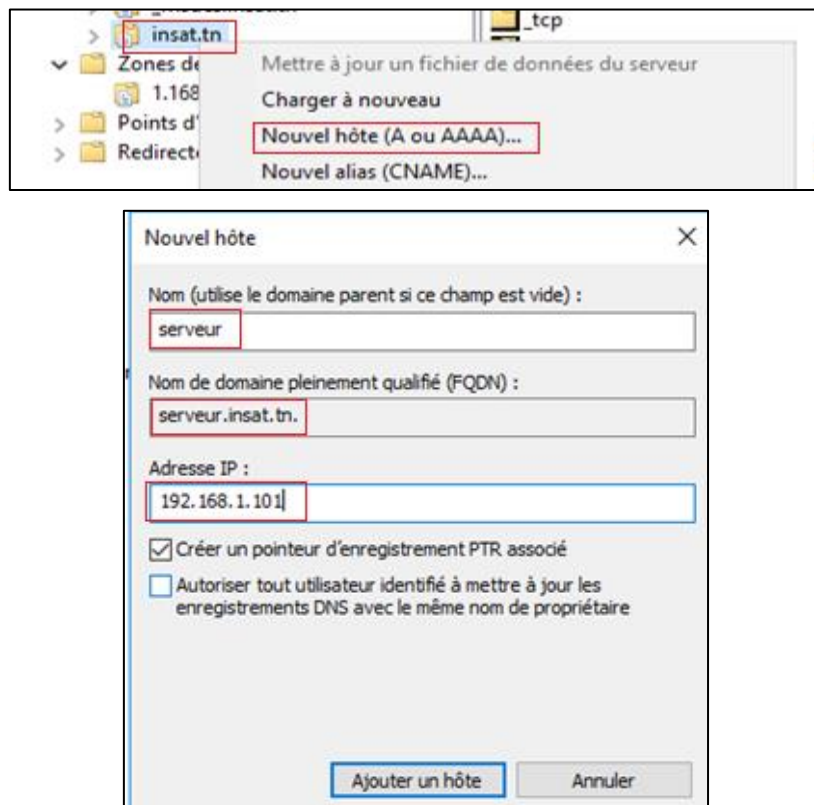


- Ajouter les hôtes :

Par défaut kdc.insat.tn existe, on va l'ajouter à la zone de recherche inversée :



Ajouter serveur et client



Actualiser, on aura maintenant :

- Zones de recherche directes (Forward) :

client	Hôte (A)	192.168.1.102	statique
kdc	Hôte (A)	192.168.1.100	statique
serveur	Hôte (A)	192.168.1.101	statique

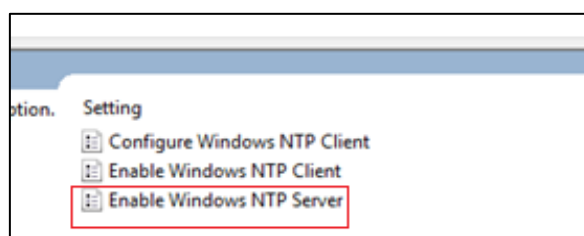
- Zones de recherche inversées (Reverse) :

192.168.1.100	Pointeur (PTR)	kdc.insat.tn.	statique
192.168.1.101	Pointeur (PTR)	serveur.insat.tn.	statique
192.168.1.102	Pointeur (PTR)	client.insat.tn.	statique

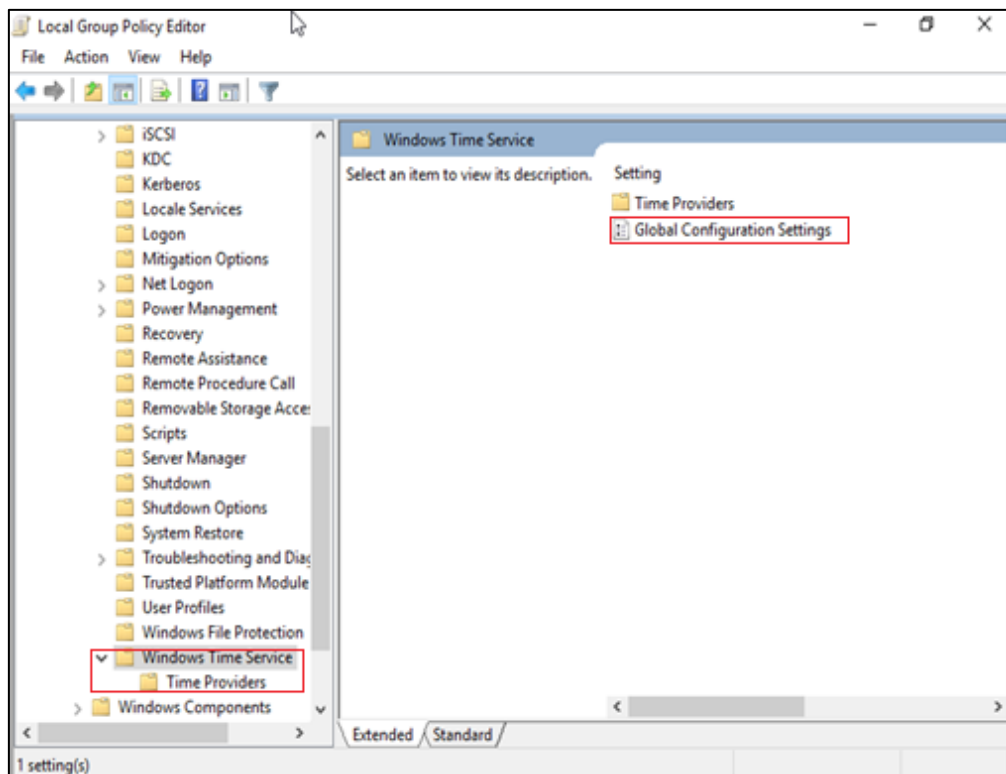
1.5- Installation et configuration du serveur NTP

Nous allons tout d'abord configurer le NTP via les stratégies de groupe en local. Tout d'abord on ouvre la commande Run (Windows + R) et on tape : **gpedit.msc**

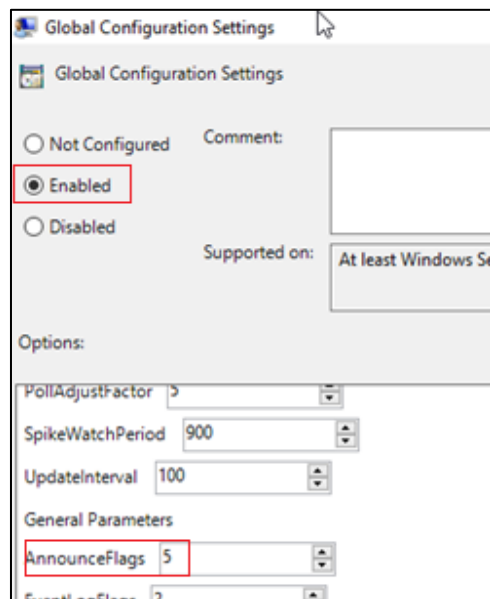
On sélectionne dans configuration de l'ordinateur : "Modèle d'administration\System\Service de temps Windows\Fournisseurs de temps" sur le panneau de gauche et on ouvre Enable Windows NTP Server :



Revenir en arrière au niveau du service de temps Windows et ouvrir les options de configuration globale :



On active la fonctionnalité et on change la valeur du paramètre « AnnounceFlag » à 5.



Ouvrez l'onglet Services sous outils :

Dans l'onglet services démarrer le Service Temps Windows et démarrer ou redémarrer le si il est déjà actif. De plus configuré son démarrage en automatique si cette option n'est pas en place par défaut.

this service is disabled, any services that explicitly depend on it will fail to start.	Windows Remote Manage...	Windows R...	Running	Automatic
	Windows Search	Provides co...	Disabled	Disabled
	Windows Time	Maintains d...	Running	Automatic (T...
	Windows Update	Enables the ...	Running	Manual (Trig...

Remarque : Si le pare-feu Windows est activé, autorisez le port 123 en UDP

2. Configuration de la machine Serveur

2.1- Configuration d'une IP statique et du nom de la machine :

ip link show (pour identifier le nom de votre interface)

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc
  group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP
  de DEFAULT group default qlen 1000
    link/ether 08:00:27:7e:81:55 brd ff:ff:
```

nano /etc/networking/interfaces

(ajouter la configuration d'une adresse IP statique avec le nom de votre interface)

```
auto lo
iface lo inet loopback
auto enp0s3
iface enp0s3 inet static
    address 192.168.1.101
    netmask 255.255.255.0
    gateway 192.168.1.1
```

sudo hostnamectl set-hostname serveur

(choisir un nouveau pour votre machine , ce nom va apparaitre après dans Active Directory)

nano /etc/hosts

(changer toute occurrence de l'ancien nom avec le nouveau nom)

```
127.0.0.1    localhost
127.0.1.1    serveur
```

Redémarrer votre machine

2.2- Configuration du DNS

nano /etc/resolv.conf

(spécifier l'adresse de votre serveur DNS)

```
nameserver 192.168.1.100
search insat.tn
```

nslookup

(Vérifier votre DNS , la résolution doit être réussite dans les deux sens)

```
> 192.168.1.100
100.1.168.192.in-addr.arpa    name = kdc.insat.tn.
> 192.168.1.101
101.1.168.192.in-addr.arpa    name = serveur.insat.tn.
> 192.168.1.102
102.1.168.192.in-addr.arpa    name = client.insat.tn.
> kdc.insat.tn
Server:      192.168.1.100
Address:     192.168.1.100#53

Name:   kdc.insat.tn
Address: 192.168.1.100
> serveur.insat.tn
Server:      192.168.1.100
Address:     192.168.1.100#53

Name:   serveur.insat.tn
Address: 192.168.1.101
> client.insat.tn
Server:      192.168.1.100
Address:     192.168.1.100#53

Name:   client.insat.tn
Address: 192.168.1.102
```

2.3- Configuration NTP

`nano /etc/ntp.conf`

(commenter les serveurs NTP par défaut et activer votre nouveau serveur NTP)

```
#pool 0.ubuntu.pool.ntp.org iburst
#pool 1.ubuntu.pool.ntp.org iburst
#pool 2.ubuntu.pool.ntp.org iburst
#pool 3.ubuntu.pool.ntp.org iburst

# Use Ubuntu's ntp server as a fallback
#pool ntp.ubuntu.com
server 192.168.1.100
server obelix
# Access control configuration: see /usr/share/doc/ntp-doc/html/
```

`service ntp restart`

`systemctl enable ntp`

`ntpq -p`

(vérifier votre serveur NTP)

remote	refid	st	t	when	poll	reach	delay	offset	jitter
kdc.insat.tn	.LOCL.	1	u	20	64	1	0.316	-363.66	0.000

`ntpdate -dv 192.168.1.100`

(vous devez avoir le message "adjust time server")

```
30 Apr 06:27:12 ntpdate[4094]: ntpdate 4.2.8p10@1.3728-o (1)
Looking for host 192.168.1.100 and service ntp
192.168.1.100 reversed to kdc.insat.tn
host found : kdc.insat.tn
transmit(192.168.1.100)
receive(192.168.1.100)
transmit(192.168.1.100)
receive(192.168.1.100)
transmit(192.168.1.100)
receive(192.168.1.100)
transmit(192.168.1.100)
receive(192.168.1.100)
server 192.168.1.100, port 123
stratum 1, precision -23, leap 00, trust 000
refid [LOCL], delay 0.02611, dispersion 0.00038
transmitted 4, in filter 4
reference time: e254bf8f.4767e76f Thu, Apr 30 2020 5:07:59.278
originate timestamp: e254d225.c767e76f Thu, Apr 30 2020 6:27:17.778
transmit timestamp: e254d226.24400e8e Thu, Apr 30 2020 6:27:18.141
filter delay: 0.02617 0.02612 0.02644 0.02611
0.00000 0.00000 0.00000 0.00000
filter offset: -0.36344 -0.36343 -0.36292 -0.36292
0.000000 0.000000 0.000000 0.000000
delay 0.02611, dispersion 0.00038
offset -0.362922

30 Apr 06:27:18 ntpdate[4094]: adjust time server 192.168.1.100 offset -0.362922 sec
```

2.4- Configuration SMB/Winbind

2.4.1- Installation des paquets nécessaires

`apt-get update`

`apt-get install winbind libpam-winbind libnss-winbind libnss3 smaba smbclient krb5-user`

2.4.2- Fichiers de configuration

- `krb5.conf`

```
sudo mv /etc/krb5.conf /etc/krb5.conf.reference
sudo nano /etc/krb5.conf
```

```
[libdefaults]
default_realm = INSAT.TN
rdns = no
dns_lookup_kdc = false
dns_lookup_realm = false
kdc_timesync = 1
ccache_type = 4
forwardable = true
proxiable = true

[realms]
INSAT.TN = {
kdc = KDC.insat.tn
admin_server = KDC.insat.tn
default_domain = insat.tn
}

[domain_realm]
.insat.tn = INSAT.TN
insat.tn = INSAT.TN

[logging]
kdc = FILE:/var/log/krb5/krb5kdc.log
admin_server = FILE:/var/log/krb5/kadmind.log
default = SYSLOG:NOTICE:DAEMON
```

- smb.conf

```
sudo mv /etc/samba/smb.conf /etc/samba/smb.conf.reference
sudo nano /etc/samba/smb.conf
```

```
#GLOBAL PARAMETERS
[global]
# [ REALM Config ]
workgroup = INSAT
realm = INSAT.TN
# Active Directory System
security = ADS
encrypt passwords = yes
# [ Logging Config ]
log level = 3
log file = /var/log/samba/%m
max log size = 50
# [ Winbind Config ]
#enable enum users and groups to display all
#domain users and groups with getent tool
#(for testing purposes only :remove for production)
winbind enum users = Yes
winbind enum groups = Yes
winbind nested groups = Yes
#to log in with username instead of username@INSAT.TN
winbind use default domain = Yes
#separator for valid users in sharing options
winbind separator = +
# [ idmap Config ]
#to map domain accounts to a local account
#specify uid and gid range
idmap uid = 600-20000
idmap gid = 600-20000
# [ User Config ]
#home directory and shell for logon users
template shell = /bin/bash
template homedir = /home/%u
```

- smb.conf

sudo nano /etc/nsswitch.conf

```
#to enable the name service switch (NSS) library to make
#domain users and groups available to the local system:
#Keep the compact entry as first source for both databases.
#This enables NSS to look up domain users and groups
#from the /etc/passwd and /etc/group files
#before querying the Winbind service.
#Do not add the winbind entry to the NSS
#shadow database. This can cause the wbinfo utility fail.
passwd:      compat systemd winbind
group:       compat systemd winbind
shadow:      compat
gshadow:     files
```

sudo service winbind restart

2.5- Configuration des dossiers partagés

Concernant la configuration des dossiers partagés sur le serveur, on a créé un dossier nommé « rt4 » contenant des sous dossiers portant le nom de chaque option : « sécurité » « datasc » « iot ».

Chaque sous dossier n'est accessible que par les étudiants appartenant au groupe de l'option spécifiée.

- smb.conf

Chaque configuration contient :

- Le chemin d'accès : par exemple //serveur/security
- Le chemin du dossier : par exemple /rt4/security
- Les utilisateurs ayant accès à ce partage : par exemple : @INSAT+security
 - on utilise le symbole « @ » pour désigner un groupe d'utilisateur suivi du nom de REALM , suivi d'un symbole « + » déjà mentionné dans la configuration globale , suivi du nom de groupe
 - pour mentionner un utilisateur , on utilise la même syntaxe en éliminant le symbole « @ »
- Les options et les permissions du partage

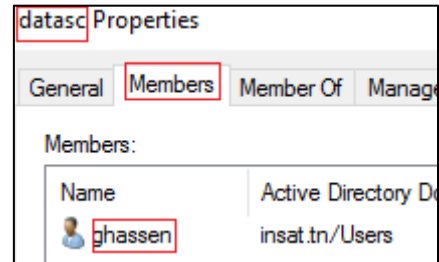
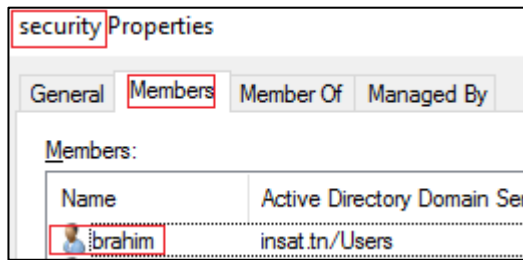
```
[security]
comment = security share
path = /rt4/security/
valid users = @INSAT+security
force group = security
writable = yes
read only = no
force create mode = 0660
create mask = 0777
directory mask = 0777
force directory mode = 0770
access based share enum = yes
hide unreadable = yes

[datasc]
comment = data science share
path = /rt4/datasc/
valid users = @INSAT+datasc
force group = datasc
writable = yes
read only = no
force create mode = 0660
create mask = 0777
directory mask = 0777
force directory mode = 0770
access based share enum = yes
hide unreadable = yes

[iot]
comment = Iot Share
path = /rt4/iot/
valid users = @INSAT+iot
force group = iot
writable = yes
read only = no
force create mode = 0660
```


- Active Directory

Maintenant, il faut créer les groupes correspondant dans AD et y ajouter des utilisateurs



3. Configuration de la machine Client

Sur la machine client, les mêmes étapes que la machine serveur vont être répétées sauf la partie du partage du dossier. Eventuellement le client peut jouer le rôle du serveur en partageant des dossiers avec smb.

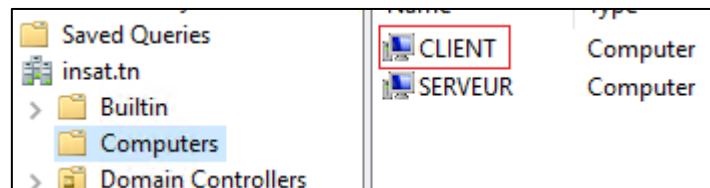
4. Authentification Kerberos

4.1- Joindre le royaume :

Ignorer le message « DNS update failed »

```
project@client:~$ sudo net ads join -U kdc -S kdc.insat.tn
Enter kdc's password:
Using short domain name -- INSAT
Joined 'CLIENT' to dns domain 'insat.tn'
DNS update failed: NT_STATUS_UNSUCCESSFUL
project@client:~$
```

Vérifier dans AD:



```
sudo systemctl winbind stop
```

```
sudo systemctl smb restart
```

```
sudo systemctl winbind start
```

Vérifier votre connexion à l'AD et lister AD users et groups

```
project@client:~$ wbinfo -u
kdc
guest
defaultaccount
krbtgt
client
serveur
```

```
project@client:~$ wbinfo -g
domain computers
domain controllers
schema admins
enterprise admins
cert publishers
domain admins
```

```
project@client:~$ sudo net ads testjoin
Join is OK
```

Par exemple l'utilisateur "serveur" n'existe pas dans /etc/passwd mais existe dans AD

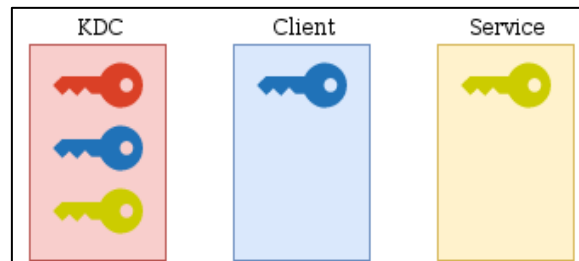
```
project@client:~$ sudo cat /etc/passwd | grep serveur
project@client:~$ id serveur
uid=606(serveur) gid=604(domain users) groups=604(domain users)
project@client:~$
```

4.2- Obtention TGT

4.2.1- Partie théorique :

Nous sommes donc dans un contexte Active Directory, ce qui fait que le KDC est également le contrôleur de domaine (*Domain Controller* ou DC). Le KDC possède l'ensemble des informations du domaine, dont les clés de chacun des services, machines, utilisateurs, ... Tous les autres éléments ne connaissent que leur clé, et n'ont de ce ne fait pas connaissance des clés des autres objets dans l'Active Directory.

Nous sommes donc dans une situation pareille :



Le client veut communiquer avec le service CIFS du serveur. Il faudra pour cela qu'il s'authentifie auprès du KDC pour ensuite pouvoir demander d'utiliser le service. Cette phase s'appelle *Authentication Service* (AS).

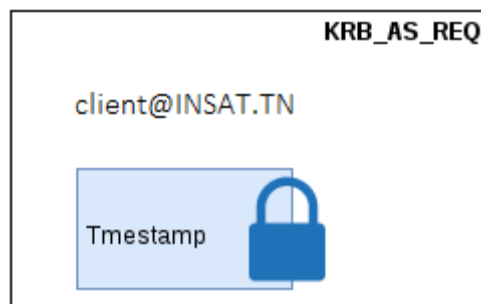
- **Authentication Service (AS)**

- **KRB_AS_REQ**

Le client va dans un premier temps envoyer une demande de *Ticket Granting Ticket* (TGT) au contrôleur de domaine (DC). Cette demande est appelée **KRB_AS_REQ** (*Kerberos Authentication Service Request*).

Le TGT que demande le client est un bout d'information chiffrée contenant entre autres une clé de session et des informations sur l'utilisateur (ID, nom, groupes, ...).

Afin d'effectuer cette demande de TGT, le client va envoyer son nom au KDC ainsi que l'heure précise de la demande (heure qu'il va chiffrer avec son secret) et quelques autres informations en clair.



Le KDC va alors recevoir ce nom, et va vérifier qu'il existe dans sa base de données.

	alice	<hash alice>
	bob	<hash bob>

Trouvé ➤	client	< hash client >

S'il le trouve, il va alors récupérer le condensat (ou *hash*) du mot de passe de client qu'il utilisera pour tenter de déchiffrer le timestamp envoyé. S'il n'y arrive pas, c'est que le client n'est pas celui qu'il prétend être.

S'il y arrive, en revanche, c'est que c'est bien client qui est en train de lui parler puisque l'utilisateur a connaissance du secret de client, donc le KDC va générer une clé de session qui sera unique pour cet utilisateur, ce ticket, et limitée dans le temps.

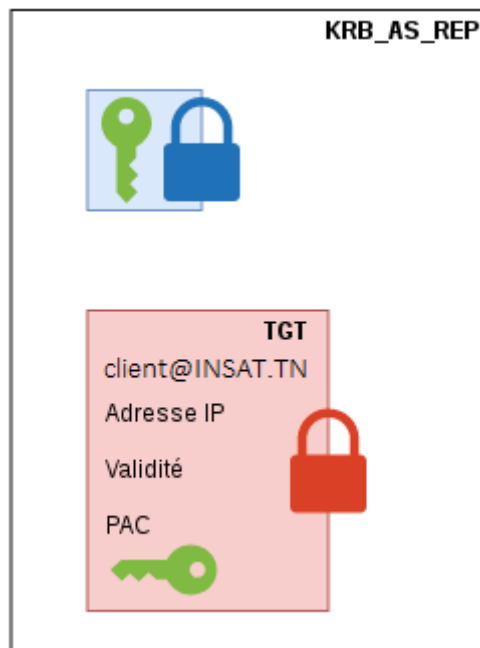


➤ KRB_AS_REP

Le KDC va alors renvoyer à client différents éléments dans sa réponse (KRB_AS_REP)

- La **clé de session**, chiffrée avec le hash de client ;
- Le **TGT**, contenant différentes informations dont les principales sont les suivantes :
 - Le nom de l'utilisateur
 - La période de validité
 - La clé de session générée
 - Le *Privilege Attribute Certificate* (PAC) qui contient des informations spécifiques sur le client permettant de connaître ses droits (son ID, les groupes auxquels il appartient, ...)

Le TGT sera chiffré avec la clé du KDC. Ainsi, seul le KDC est en mesure de déchiffrer et lire le contenu de ce ticket.



Remarque : Notons que ce TGT est considéré comme une information publique. Il peut très bien être intercepté pendant la phase d'authentification.

Le client reçoit alors ces informations. En utilisant son secret, le premier message va être déchiffré afin de récupérer la clé de session nécessaire pour la suite des échanges.

4.2.2- Partie pratique :

```
project@client:~$ kinit client
Password for client@INSAT.TN:
project@client:~$ klist
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: client@INSAT.TN

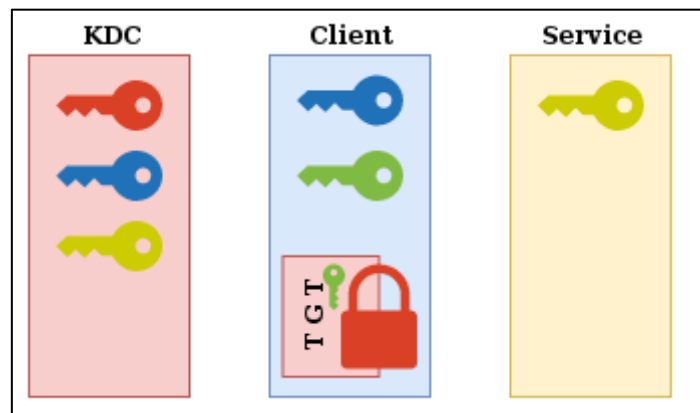
Valid starting    Expires          Service principal
05/03/20 02:32:03 05/03/20 12:32:03 krbtgt/INSAT.TN@INSAT.TN
                renew until 05/04/20 02:31:55
project@client:~$
```

4.3- Obtention TGS et accès au service

4.3.1- Partie théorique

• Ticket-Granting Service (TGS)

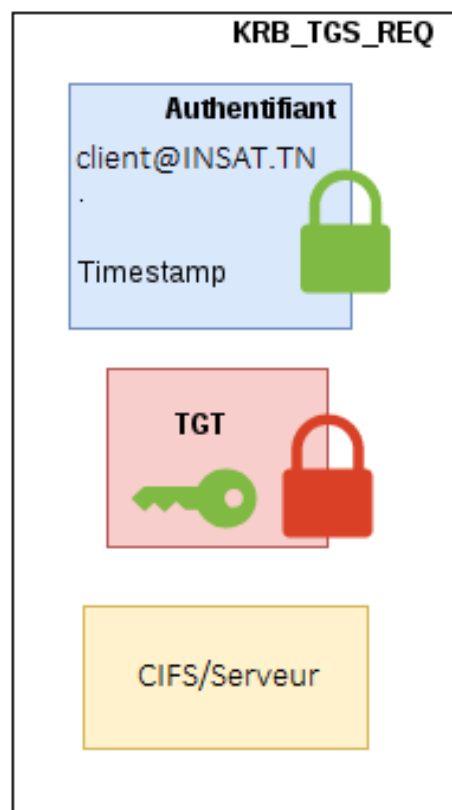
Maintenant que le client a pu s'authentifier, nous voici dans la situation suivante : Le client possède sa clé ainsi qu'une clé de session limitée dans le temps que seul lui connaît, et un TGT chiffré par le KDC qui contient, entre autres, cette même clé de session.



➤ KRB_TGS_REQ

Si le client veut utiliser un service, par exemple CIFS sur le serveur \\SERVEUR, il va envoyer plusieurs informations au KDC pour que celui-ci lui renvoie un ticket de service.

- Le TGT;
- L'identifiant du service qu'il veut utiliser et l'hôte associé, donc CIFS/SERVEUR dans cet exemple;
- Un *authenticator*, qui est un message contenant son nom, et un timestamp, le tout chiffré avec la clé de session qu'il a en sa possession.



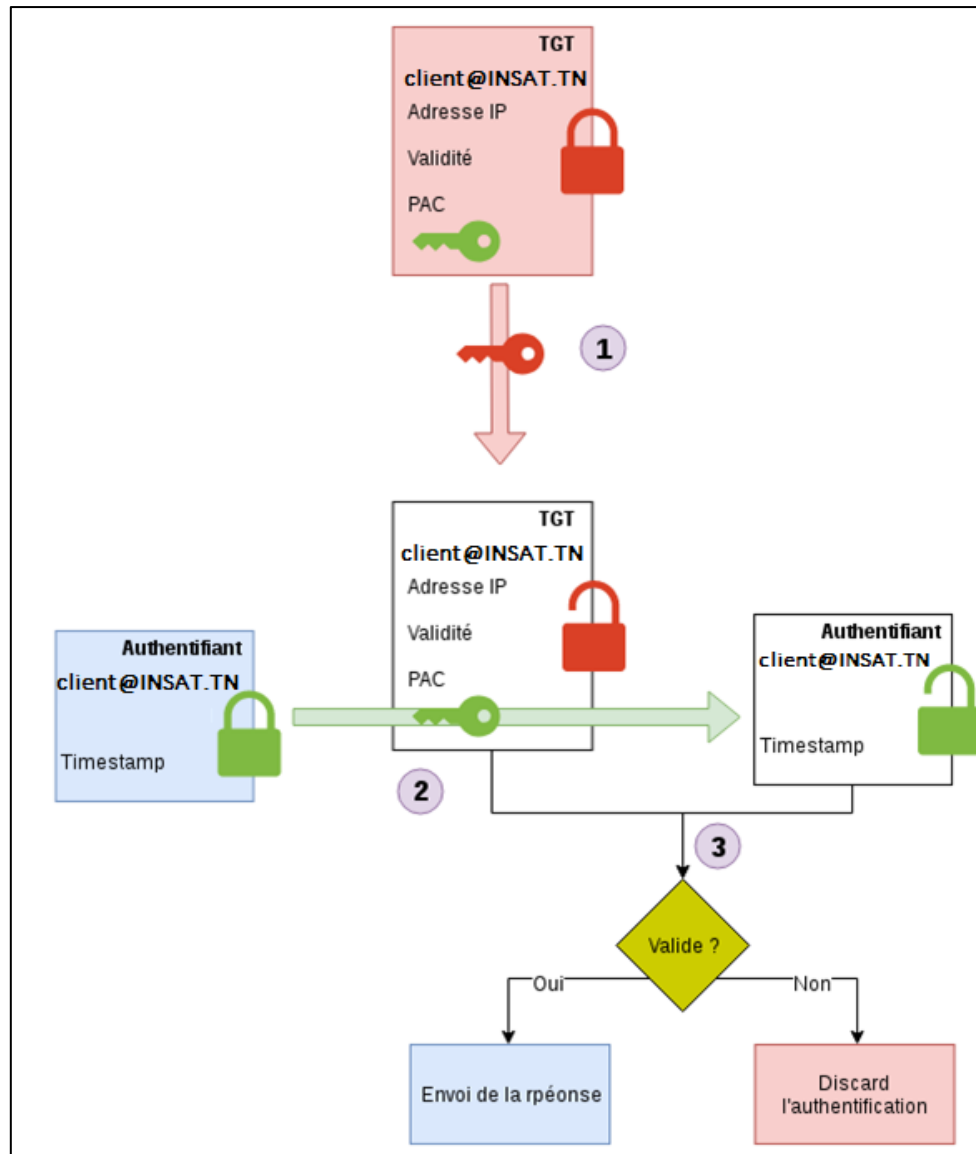
Ces informations reçues par le KDC permettent **deux choses**.

- La **première** est de s'assurer que c'est bien le client qui fait la demande. Pour cela, le KDC va comparer le contenu du TGT avec le contenu de l'*authenticator*. Comme seul le KDC peut lire le contenu du TGT, il est certain que ce contenu n'a pas été falsifié. Le KDC va donc lire le contenu du TGT, donc les

informations de l'utilisateur qui possède le TGT, mais également la clé de session. Ensuite, il va déchiffrer le contenu de l'*authenticator* avec la clé de session. Si le déchiffrement fonctionne, et que les données dans l'*authenticator* correspondent aux données dans le TGT, alors client est bien qui il prétend être. En effet, cela assure au KDC que celui qui a fait la requête possède le TGT et a connaissance de la clé de session négociée.

- La **deuxième** est de savoir à quel service client veut avoir accès, information qu'il obtient en recevant l'identifiant de ce service.

Voici un schéma qui permet de résumer ce processus de vérification au niveau du KDC :

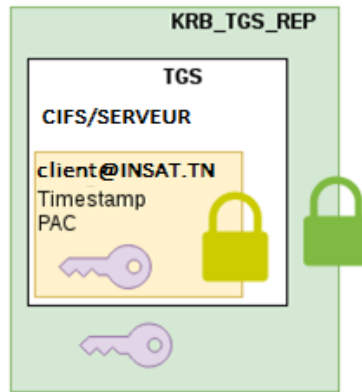


➤ KRB_TGS_REP

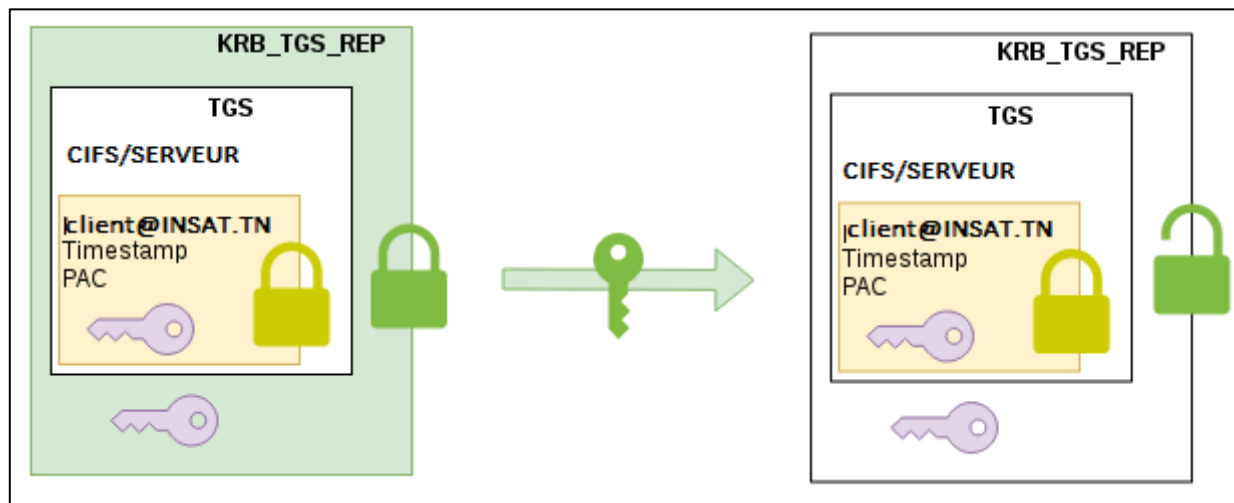
Maintenant que le KDC a pu vérifier que l'utilisateur était bien client, il va lui renvoyer des informations qui lui permettront de faire une demande auprès du service. Ce message est le **KRB_TGS_REP**. Il est composé des éléments suivants :

- Un ticket contenant le nom et l'instance du service demandé (CIFS/SERVEUR), le nom d'utilisateur (client), le PAC et une nouvelle clé de session qui est valide uniquement pour client voulant discuter avec CIFS sur //SERVEUR pendant un certain temps. Ce ticket est chiffré avec la clé du service en question (donc celle de la machine, puisque le service CIFS tourne sous l'utilisateur machine);
- La nouvelle clé de session

Ces deux informations (le ticket et la clé de session) sont chiffrées avec la première clé de session, celle qui a été échangée au début entre le KDC et le client.



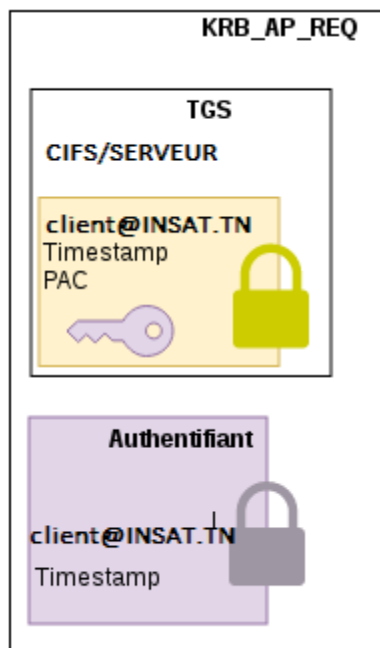
Le client va recevoir ce paquet, et va pouvoir déchiffrer la première couche pour obtenir la clé de session créée pour la communication avec le service, ainsi que le ticket généré pour l'utilisation de ce service. Ce ticket s'appelle le Ticket-Granting Service (TGS).



- **Accès au service (AP)**

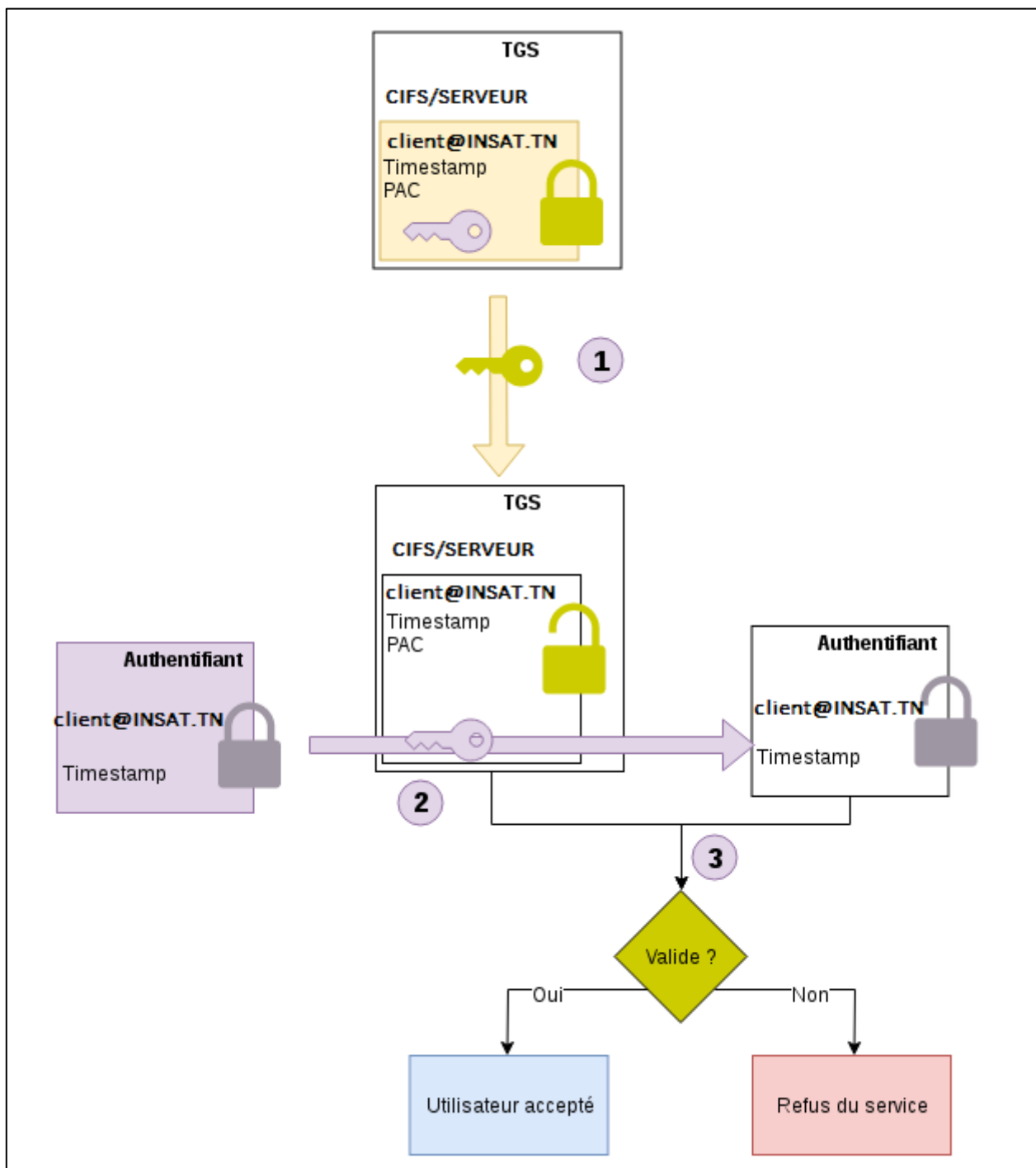
- **KRB_AP_REQ**

Le client va alors générer un nouvel authentifiant qu'il va chiffrer avec cette nouvelle clé de session, et enverra le ticket par la même occasion pour envoyer la requête **KRB_AP_REQ** au service. C'est le même processus qu'avec le KDC.



Le service CIFS reçoit le ticket qu'il peut déchiffrer. Il est certain que celui-ci est valide et authentique puisque seul le KDC est l'autre entité possédant sa clé. Dedans, il trouvera la clé de session qu'il utilisera pour déchiffrer l'authentifiant. En comparant le contenu du ticket de service avec le contenu de l'authentifiant, le service peut être certain de l'authenticité du client, et il peut lui envoyer les informations dont il a besoin.

Voici un schéma qui permet de résumer ce processus de vérification au niveau du SERVEUR :

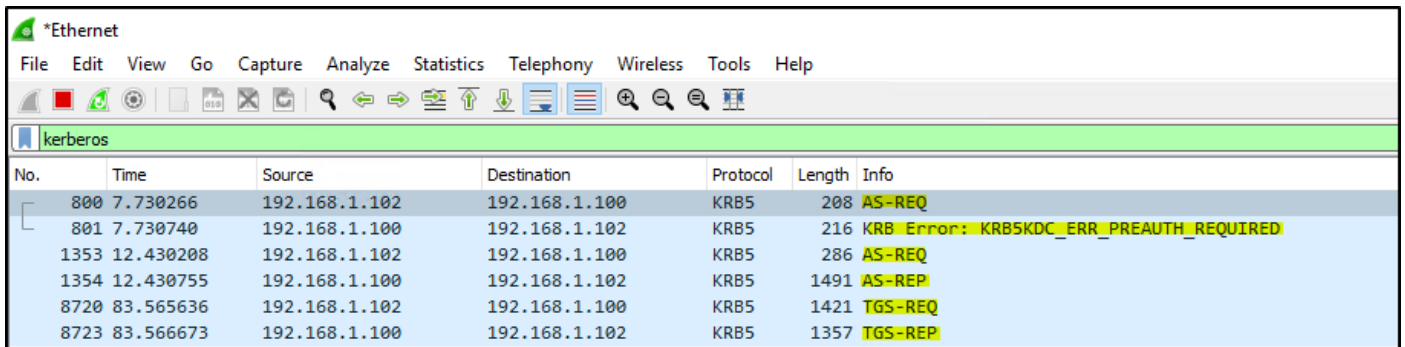


4.3.2- Partie pratique :

Toute la procédure de génération du TGS et de vérification du TGS citée ci-dessus se passe en « **Background** », de ce fait le client, après avoir obtenu son TGT, il peut demander un service. Si son TGT est valide, il va obtenir un TGS et ceci va être vérifié au niveau du serveur.

Nous avons intercepté ce processus qui s'exécute en arrière-plan grâce au **Wireshark**.

- Au niveau du KDC



The image shows a Wireshark packet capture of Kerberos traffic on an Ethernet interface. The filter is set to 'kerberos'. The packet list shows several messages between 192.168.1.102 and 192.168.1.100. The first AS-REQ is rejected (KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED), followed by a successful AS-REP, then a TGS-REQ and a TGS-REP.

No.	Time	Source	Destination	Protocol	Length	Info
800	7.730266	192.168.1.102	192.168.1.100	KRB5	208	AS-REQ
801	7.730740	192.168.1.100	192.168.1.102	KRB5	216	KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED
1353	12.430208	192.168.1.102	192.168.1.100	KRB5	286	AS-REQ
1354	12.430755	192.168.1.100	192.168.1.102	KRB5	1491	AS-REP
8720	83.565636	192.168.1.102	192.168.1.100	KRB5	1421	TGS-REQ
8723	83.566673	192.168.1.100	192.168.1.102	KRB5	1357	TGS-REP

AS-REQ : Authentication Service Request, c'est la demande de TGT : En premier lieu elle a été refusé, puis accepté et suivie de renvoi de AS-REP (réponse du KDC et renvoi du TGT)

TGS-REQ : Ticket-Granting Service Request, une fois le client un service et son TGT et valide, il recevra un TGS-REP

- Au niveau du client : Exemple 1

l'option **-k** de la commande **smbclient** désigne : **Use kerberos (active directory) authentication**

```
client@client:~$ klist
klist: No credentials cache found (filename: /tmp/krb5cc_1002)
client@client:~$ smbclient //serveur/security -k
WARNING: The "idmap uid" option is deprecated
WARNING: The "idmap gid" option is deprecated
SPNEGO: Could not find a suitable mechtype in NEG_TOKEN_INIT
session setup failed: NT_STATUS_INTERNAL_ERROR
client@client:~$ kinit
Password for client@INSAT.TN:
client@client:~$ klist
Ticket cache: FILE:/tmp/krb5cc_1002
Default principal: client@INSAT.TN

Valid starting    Expires          Service principal
05/03/20 23:05:25 05/04/20 09:05:25 krbtgt/INSAT.TN@INSAT.TN
renew until 05/04/20 23:05:20
client@client:~$ smbclient //serveur/security -k
WARNING: The "idmap uid" option is deprecated
WARNING: The "idmap gid" option is deprecated
Try "help" to get a list of possible commands.
smb: \> ls
```

Au début, le client ne peut pas accéder au service car il n'a pas de TGT, ensuite, on remarque que l'accès est devenu possible avec obtention d'une TGS (CIFS) **automatiquement**.

```

client@client:~$ klist
Ticket cache: FILE:/tmp/krb5cc_1002
Default principal: client@INSAT.TN

Valid starting    Expires          Service principal
05/03/20 23:05:25 05/04/20 09:05:25 krbtgt/INSAT.TN@INSAT.TN
    renew until 05/04/20 23:05:20
05/03/20 23:05:31 05/04/20 09:05:25 cifs/serveur@INSAT.TN
client@client:~$ kdestroy
client@client:~$ klist
klist: No credentials cache found (filename: /tmp/krb5cc_1002)
client@client:~$ smbclient //serveur/security -k
WARNING: The "idmap uid" option is deprecated
WARNING: The "idmap gid" option is deprecated
SPNEGO: Could not find a suitable mechtype in NEG_TOKEN_INIT
session setup failed: NT_STATUS_INTERNAL_ERROR
client@client:~$

```

Lorsque on supprime notre ticket, l'accès redevient bien sur refusé.

- Au niveau du client : Exemple 2

Maintenant , on va mettre l'accent sur l'obtention d'une TGS avec un accès refusé de la part du serveur.

```

project@client:~$ klist
klist: No credentials cache found (filename: /tmp/krb5cc_1000)
project@client:~$ kinit ghassen
Password for ghassen@INSAT.TN:
project@client:~$ klist
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: ghassen@INSAT.TN

Valid starting    Expires          Service principal
05/04/20 01:56:54 05/04/20 11:56:54 krbtgt/INSAT.TN@INSAT.TN
    renew until 05/05/20 01:56:50
project@client:~$ smbclient //serveur/security -k
WARNING: The "idmap uid" option is deprecated
WARNING: The "idmap gid" option is deprecated
tree connect failed: NT_STATUS_ACCESS_DENIED
project@client:~$ klist
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: ghassen@INSAT.TN

Valid starting    Expires          Service principal
05/04/20 01:56:54 05/04/20 11:56:54 krbtgt/INSAT.TN@INSAT.TN
    renew until 05/05/20 01:56:50
05/04/20 01:57:03 05/04/20 11:56:54 cifs/serveur@INSAT.TN
project@client:~$ smbclient //serveur/datasc -k
WARNING: The "idmap uid" option is deprecated
WARNING: The "idmap gid" option is deprecated
Try "help" to get a list of possible commands.
smb: \> quit
project@client:~$ smbclient //serveur/security -k
WARNING: The "idmap uid" option is deprecated
WARNING: The "idmap gid" option is deprecated
tree connect failed: NT_STATUS_ACCESS_DENIED

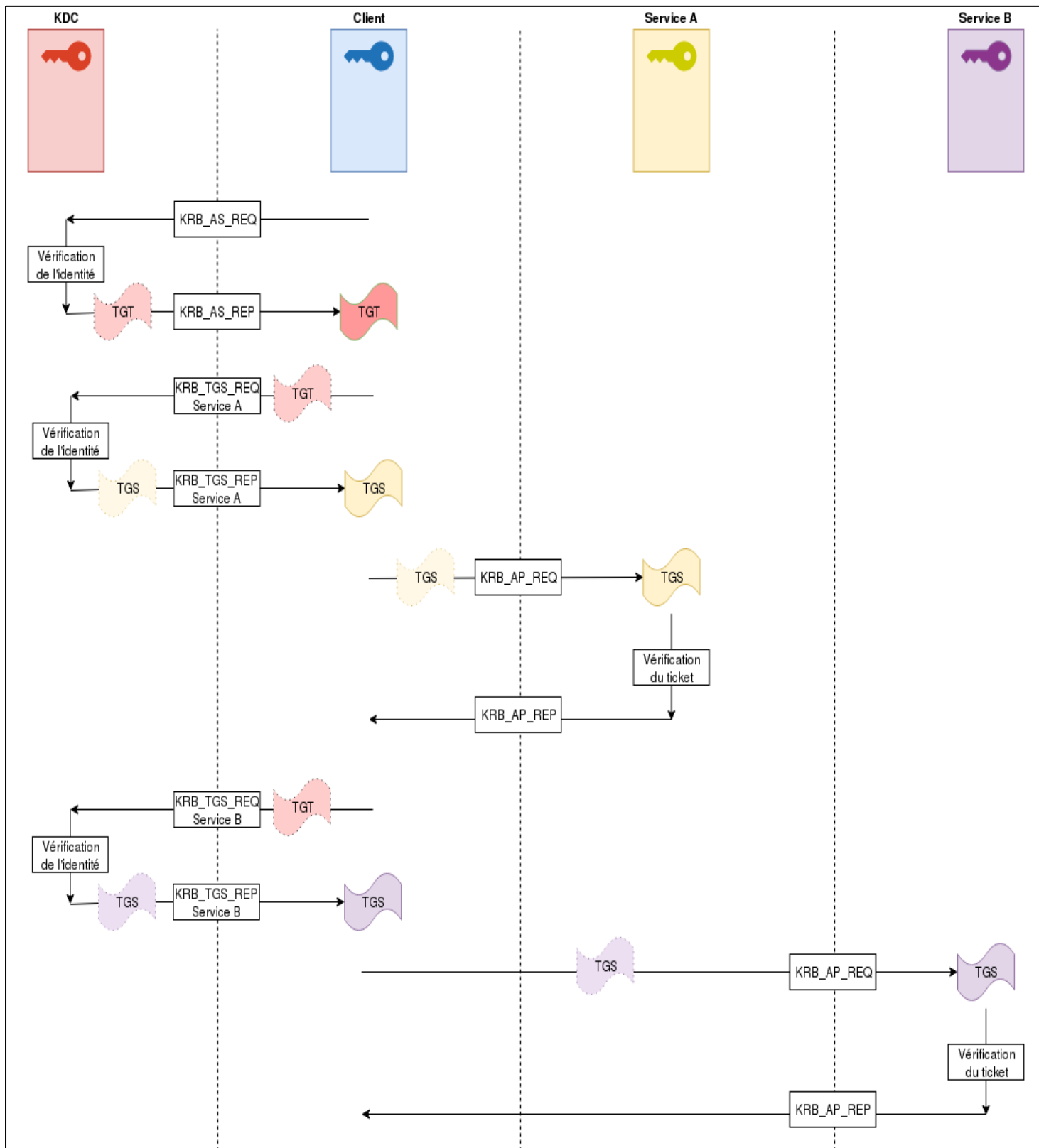
```

Au début , l'utilisateur « ghassen » appartenant au groupe « datasc » et avec une TGT valide veut accéder au dossier « security » , on remarque qu'il a obtenu automatiquement un ticket CIFS après sa demande d'accès mais son accès est refusé. Avec ce même ticket CIFS , il a pu au ensuite d'accéder au dossier « datasc » mais son accès au dossier « security » reste bloqué.

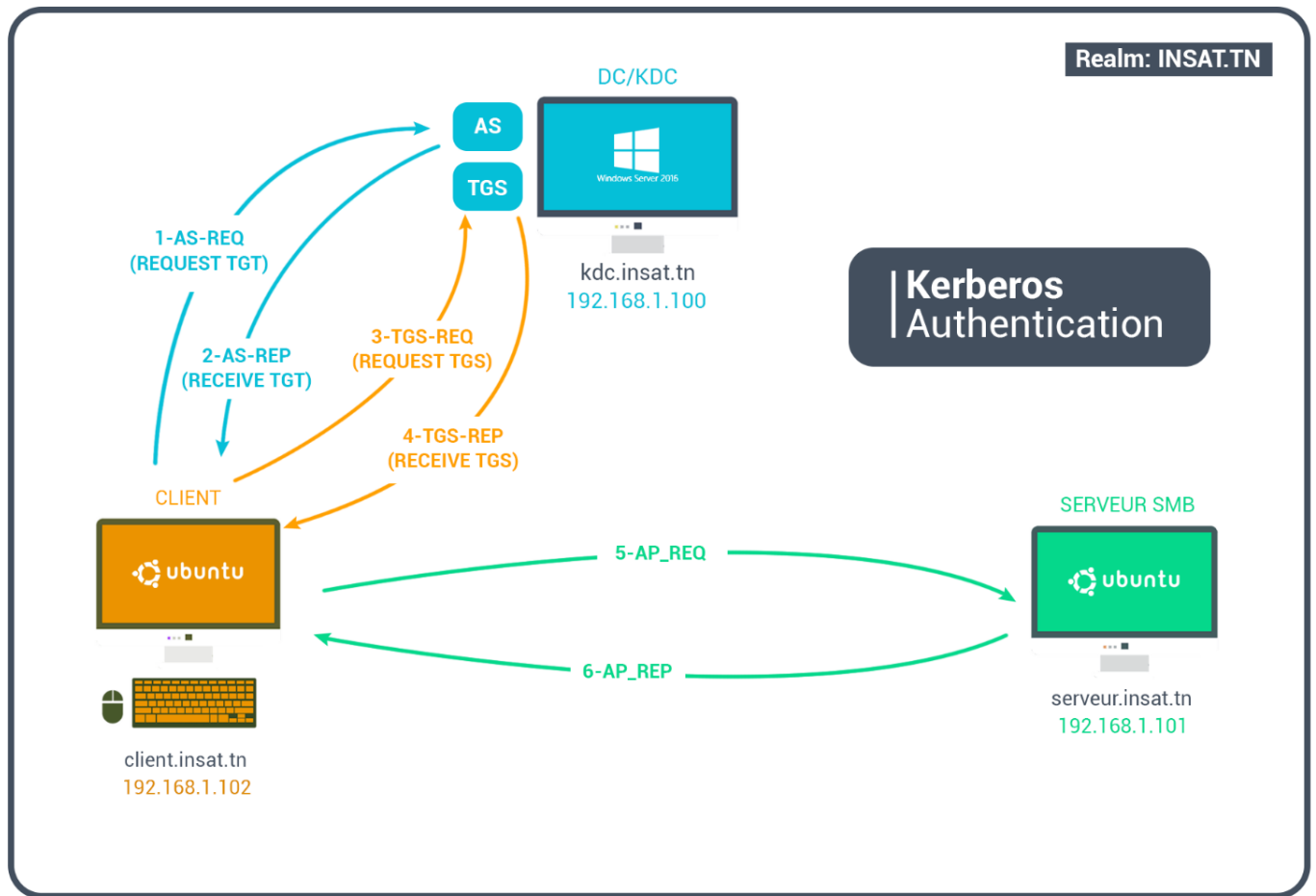
4.4- Résumé

C'est un processus relativement complexe, mais une fois que les étapes ont été vues en détails, on comprend mieux l'utilité de chacune d'elles.

Voici un schéma récapitulatif des trois étapes pour un client qui demande un accès à deux services différents :



Pour résumer notre projet on peut se référer à ce schéma :



IV- Conclusion générale

Ce projet nous a permis de découvrir les étapes détaillées de mise en œuvre du protocole d'authentification Kerberos, d'analyser son processus d'authentification et de connecter des machines linux dans un domaine Windows (Active Directory).

Kerberos, en tant que système d'authentification, n'est qu'un des maillons indissociables de toute la chaîne de sécurité. Son rôle est primordial, car il va permettre la certification de la validité des interlocuteurs sur le réseau, mais sera totalement inutile si des portes sont laissées grandes ouvertes aux crackers, leur permettant de prendre possession des comptes avec privilège.