NFSv4
Présentation

Laurent Azema
Benoit Métrot

CNRS-ICJ

ANF Les systèmes d'authentification dans la communauté ESR : étude, mise en œuvre et interfaçage dans un laboratoire de Mathématique
Angers, 22 - 26 septembre 2014

Plan

Progression

Network File System

- Système de fichiers gérant l'accès partagé à un système de fichiers sur un serveur à travers un réseau de données
- Un protocole qui évolue
  - origine SUN 1984 puis normalisation IETF
  - v2 RFC1094 Mars 1989
  - v3 RFC1813 Juin 1995
  - v4 RFC3530 Avril 2003
  - v4.1 RFC5661 Janvier 2010

Network File System

- Système de fichiers gérant l'accès partagé à un système de fichiers sur un serveur à travers un réseau de données
- Un protocole qui évolue
  - origine SUN 1984 puis normalisation IETF
  - v2 RFC1094 Mars 1989
  - v3 RFC1813 Juin 1995
  - v4 RFC3530 Avril 2003
  - v4.1 RFC5661 Janvier 2010

Objectifs du protocole

- Architecture résistante aux interruptions
- Indépendance
  - protocole réseau
  - systèmes d'exploitation
  - systèmes de fichiers
- Simplicité : opérations réduites au minimum utile
- Bonnes performances

Objectifs du protocole

- Architecture résistante aux interruptions
- Indépendance
  - protocole réseau
  - systèmes d'exploitation
  - systèmes de fichiers
- Simplicité : opérations réduites au minimum utile
- Bonnes performances

Objectifs du protocole

- Architecture résistante aux interruptions
- Indépendance
  - protocole réseau
  - systèmes d'exploitation
  - systèmes de fichiers
- Simplicité : opérations réduites au minimum utile
- Bonnes performances

Objectifs du protocole

- Architecture résistante aux interruptions
- Indépendance
  - protocole réseau
  - systèmes d'exploitation
  - systèmes de fichiers
- Simplicité : opérations réduites au minimum utile
- Bonnes performances

Quelques principes de fonctionnement

- Utilisation du modèle OSI
  - Remote Procedure Call (RFC1831)
  - eXternal Data Representation (RFC1832)
- Abstraction du système de fichiers réel du serveur
  - Serveur traduit requête client en commandes fs réel
  - fonctions : NFS client ∩ NFS serveur ∩ fs serveur
- Cache local sur le client : fichiers, attributs, répertoires
- Compromis perf/sûreté ; ex. modes sync/async

Quelques principes de fonctionnement

- Utilisation du modèle OSI
  - Remote Procedure Call (RFC1831)
  - eXternal Data Representation (RFC1832)
- Abstraction du système de fichiers réel du serveur
  - Serveur traduit requête client en commandes fs réel
  - fonctions : NFS client ∩ NFS serveur ∩ fs serveur
- Cache local sur le client : fichiers, attributs, répertoires
- Compromis perf/sûreté ; ex. modes sync/async

Quelques principes de fonctionnement

- Utilisation du modèle OSI
  - Remote Procedure Call (RFC1831)
  - eXternal Data Representation (RFC1832)
- Abstraction du système de fichiers réel du serveur
  - Serveur traduit requête client en commandes fs réel
  - fonctions : NFS client ∩ NFS serveur ∩ fs serveur
- Cache local sur le client : fichiers, attributs, répertoires
- Compromis perf/sûreté ; ex. modes sync/async

Quelques principes de fonctionnement

- Utilisation du modèle OSI
  - Remote Procedure Call (RFC1831)
  - eXternal Data Representation (RFC1832)
- Abstraction du système de fichiers réel du serveur
  - Serveur traduit requête client en commandes fs réel
  - fonctions : NFS client $\cap$ NFS serveur $\cap$ fs serveur
- Cache local sur le client : fichiers, attributs, répertoires
- Compromis perf/sûreté ; ex. modes sync/async

Progression

Une nouvelle version NFSv4

- NFSv4 est une réécriture du protocole
- Nouveautés :
  - Améliorer accès et perf. au dessus d'Internet
  - Sécurité basée sur l'identité des utilisateurs
  - Avoir une bonne interopérabilité entre les plate-formes
  - Conception compatible avec extensions normalisées
  - Optimisation des communications
  - Support redondance et migration système de fichiers
  - internationalisation UTF-8 pour noms fichiers

Une nouvelle version NFSv4

- NFSv4 est une réécriture du protocole
- Nouveautés :
  - Améliorer accès et perf. au dessus d'Internet
    - franchir les pare-feu : tcp/2049 (plus de portmap)
    - réseaux à forte latence et débit faible
    - supporter l'augmentation du nombre de clients
  - Sécurité basée sur l'identité des utilisateurs
  - Avoir une bonne interopérabilité entre les plate-formes
  - Conception compatible avec extensions normalisées
  - Optimisation des communications
  - Support redondance et migration système de fichiers
  - internationalisation UTF-8 pour noms fichiers

Une nouvelle version NFSv4

- NFSv4 est une réécriture du protocole
- Nouveautés :
  - Améliorer accès et perf. au dessus d'Internet
  - Sécurité basée sur l'identité des utilisateurs
  - Avoir une bonne interopérabilité entre les plate-formes
  - Conception compatible avec extensions normalisées
  - Optimisation des communications
  - Support redondance et migration système de fichiers
  - internationalisation UTF-8 pour noms fichiers

Une nouvelle version NFSv4

- NFSv4 est une réécriture du protocole
- Nouveautés :
  - Améliorer accès et perf. au dessus d'Internet
  - Sécurité basée sur l'identité des utilisateurs
  - Avoir une bonne interopérabilité entre les plate-formes
    - ensemble de fonctionnalités utiles et courantes
    - 3 catégories d'attributs : obligatoire, recommandé, défini
    - exemple attribut recommandé : les ACL de WindowsNT
  - Conception compatible avec extensions normalisées
  - Optimisation des communications
  - Support redondance et migration système de fichiers
  - internationalisation UTF-8 pour noms fichiers

Une nouvelle version NFSv4

- NFSv4 est une réécriture du protocole
- Nouveautés :
  - Améliorer accès et perf. au dessus d'Internet
  - Sécurité basée sur l'identité des utilisateurs
  - Avoir une bonne interopérabilité entre les plate-formes
  - Conception compatible avec extensions normalisées
  - Optimisation des communications
  - Support redondance et migration système de fichiers
  - internationalisation UTF-8 pour noms fichiers

Une nouvelle version NFSv4

- NFSv4 est une réécriture du protocole
- Nouveautés :
  - Améliorer accès et perf. au dessus d'Internet
  - Sécurité basée sur l'identité des utilisateurs
  - Avoir une bonne interopérabilité entre les plate-formes
  - Conception compatible avec extensions normalisées
  - Optimisation des communications
    - regroupement d'opérations dans une même requête RPC
    - réduction taille opérations : current filehandle et saved filehandle
  - Support redondance et migration système de fichiers
  - internationalisation UTF-8 pour noms fichiers

Une nouvelle version NFSv4

- NFSv4 est une réécriture du protocole
- Nouveautés :
  - Améliorer accès et perf. au dessus d'Internet
  - Sécurité basée sur l'identité des utilisateurs
  - Avoir une bonne interopérabilité entre les plate-formes
  - Conception compatible avec extensions normalisées
  - Optimisation des communications
  - Support redondance et migration système de fichiers
    - serveur peut informer les clients de la localisation des fs
  - internationalisation UTF-8 pour noms fichiers

Une nouvelle version NFSv4

- NFSv4 est une réécriture du protocole
- Nouveautés :
  - Améliorer accès et perf. au dessus d'Internet
  - Sécurité basée sur l'identité des utilisateurs
  - Avoir une bonne interopérabilité entre les plate-formes
  - Conception compatible avec extensions normalisées
  - Optimisation des communications
  - Support redondance et migration système de fichiers
  - internationalisation UTF-8 pour noms fichiers

Des daemon intégrés

Espace de nommage des exportations

- confié à MOUNT dans NFSv3 qui disparaît en v4
  partage = arbre à l'intérieur d'un fs
- fsid=0 ou root avec des pseudo systèmes de fichiers
- adressable directement par les clients public filehandle

Verrouillage des fichiers

- NFSv3 sans état => délègue Network Lock Manager
- rpc.lockd gère les verrous sur le serveur
- rpc.statd avertir les clients d'un redémarrage
- NFSv4 avec état ; système à base de baux
- accès client ressource serveur => renouveau des baux
- Nouveauté : la délégation => travail sur cache local

Des daemon intégrés

Espace de nommage des exportations

- confié à MOUNT dans NFSv3 qui disparaît en v4
  partage = arbre à l'intérieur d'un fs
- fsid=0 ou root avec des pseudo systèmes de fichiers
- adressable directement par les clients public filehandle

Verrouillage des fichiers

- NFSv3 sans état => délègue Network Lock Manager
- rpc.lockd gère les verrous sur le serveur
- rpc.statd avertir les clients d'un redémarrage
- NFSv4 avec état ; système à base de baux
- accès client ressource serveur => renouveau des baux
- Nouveauté : la délégation => travail sur cache local

Des daemon intégrés

Espace de nommage des exportations

- confié à MOUNT dans NFSv3 qui disparaît en v4
  partage = arbre à l'intérieur d'un fs
- fsid=0 ou root avec des pseudo systèmes de fichiers
- adressable directement par les clients public filehandle

Verrouillage des fichiers

- NFSv3 sans état => délègue Network Lock Manager
- rpc.lockd gère les verrous sur le serveur
- rpc.statd avertir les clients d'un redémarrage
- NFSv4 avec état ; système à base de baux
- accès client ressource serveur => renouveau des baux
- Nouveauté : la délégation => travail sur cache local

Des daemon intégrés

Espace de nommage des exportations

- confié à MOUNT dans NFSv3 qui disparaît en v4
  partage = arbre à l'intérieur d'un fs
- fsid=0 ou root avec des pseudo systèmes de fichiers
- adressable directement par les clients public filehandle

Verrouillage des fichiers

- NFSv3 sans état => délègue Network Lock Manager
- rpc.lockd gère les verrous sur le serveur
- rpc.statd avertir les clients d'un redémarrage
- NFSv4 avec état ; système à base de baux
- accès client ressource serveur => renouveau des baux
- Nouveauté : la délégation => travail sur cache local

Correspondance identités entre client serveur

- NFSv3 : MOUNT transmet uid/gid entre client et serveur => utilisation même uid/gid
- NFSv4 : utilise rpc.idmapd pour Id Mapping
  - noms utilisateur@domaine <=> uid/gid
  - processus local que ce soit client ou serveur
- résolution avec rpc.idmapd
  1. système de cache facultatif : request-key
     noms NFSv4 => clés de type id_resolver
  2. service local de résolution identifiant : rpc.idmapd
  3. configuration NSS par exemple avec LDAP pour voir tous les utilisateurs du domaine

Correspondance identités entre client serveur

- NFSv3 : MOUNT transmet uid/gid entre client et serveur => utilisation même uid/gid
- NFSv4 : utilise rpc.idmapd pour Id Mapping
  - noms utilisateur@domaine <=> uid/gid
  - processus local que ce soit client ou serveur
- résolution avec rpc.idmapd
  1. système de cache facultatif : request-key
     noms NFSv4 => clés de type id_resolver
  2. service local de résolution identifiant : rpc.idmapd
  3. configuration NSS par exemple avec LDAP pour voir tous les utilisateurs du domaine

Correspondance identités entre client serveur

- NFSv3 : MOUNT transmet uid/gid entre client et serveur => utilisation même uid/gid
- NFSv4 : utilise rpc.idmapd pour Id Mapping
  - noms utilisateur@domaine <=> uid/gid
  - processus local que ce soit client ou serveur
- résolution avec rpc.idmapd
  1. système de cache facultatif : request-key
     noms NFSv4 => clés de type id_resolver
  2. service local de résolution identifiant : rpc.idmapd
  3. configuration NSS par exemple avec LDAP pour voir tous les utilisateurs du domaine

fichiers de configuration

/etc/request-key.conf

```
create id_resolver * * /usr/sbin/nfsidmap %k %d
```

/etc/idmapd.conf

```
[General]
Domain = <domaineDNS>
#Local-Realms = <DOMAIN>

[Mapping]
Nobody-User = nobody
Nobody-Group = nobody

[Translation]
Method = nsswitch
```

Progression

Méthodes de sécurité

- sys : basée sur @IP client = par défaut en NFSv3
- kerberos avec RPCSEC_GSS (RFC2203)
- Authentification : principal utilisateur
  Accounting : IdMapping
  Autorisation : droits sur le fs réel
- Services de sécurité :
  - krb5 : authentification seule
  - krb5i : intégrité des données
  - krb5p : confidentialité des données
- Identification client avec principal nfs/hostname
  => connexion inverse pour délégation

Méthodes de sécurité

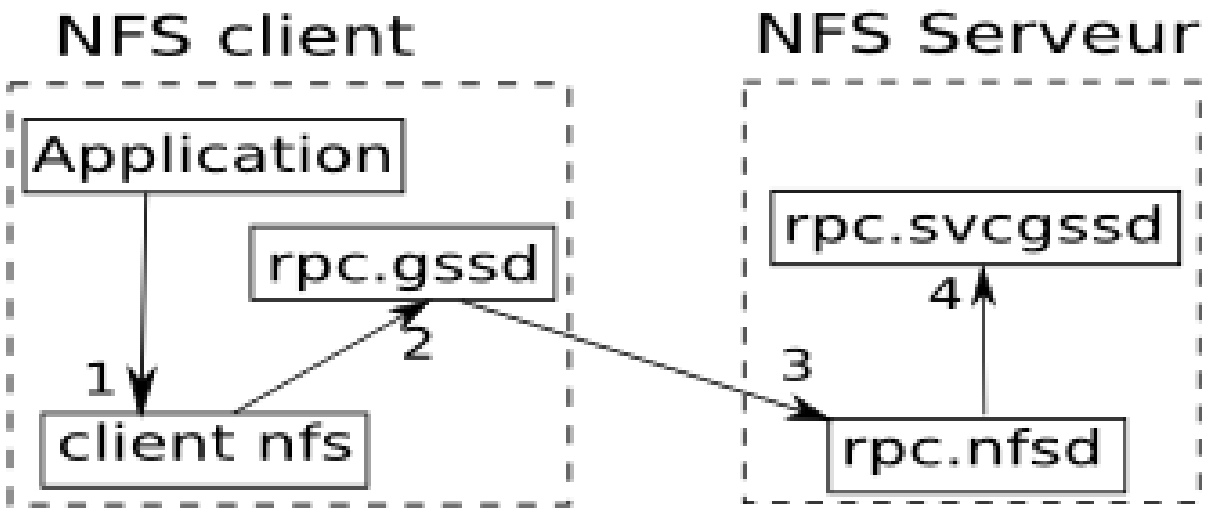
- sys : basée sur @IP client = par défaut en NFSv3
- kerberos avec RPCSEC_GSS (RFC2203)
- Authentification : principal utilisateur
  Accounting : IdMapping
  Autorisation : droits sur le fs réel
- Services de sécurité :
  - krb5 : authentification seule
  - krb5i : intégrité des données
  - krb5p : confidentialité des données
- Identification client avec principal nfs/hostname
  => connexion inverse pour délégation

Méthodes de sécurité

- sys : basée sur @IP client = par défaut en NFSv3
- kerberos avec RPCSEC_GSS (RFC2203)
- Authentification : principal utilisateur
  Accounting : IdMapping
  Autorisation : droits sur le fs réel
- Services de sécurité :
  - krb5 : authentification seule
  - krb5i : intégrité des données
  - krb5p : confidentialité des données
- Identification client avec principal nfs/hostname
  => connexion inverse pour délégation

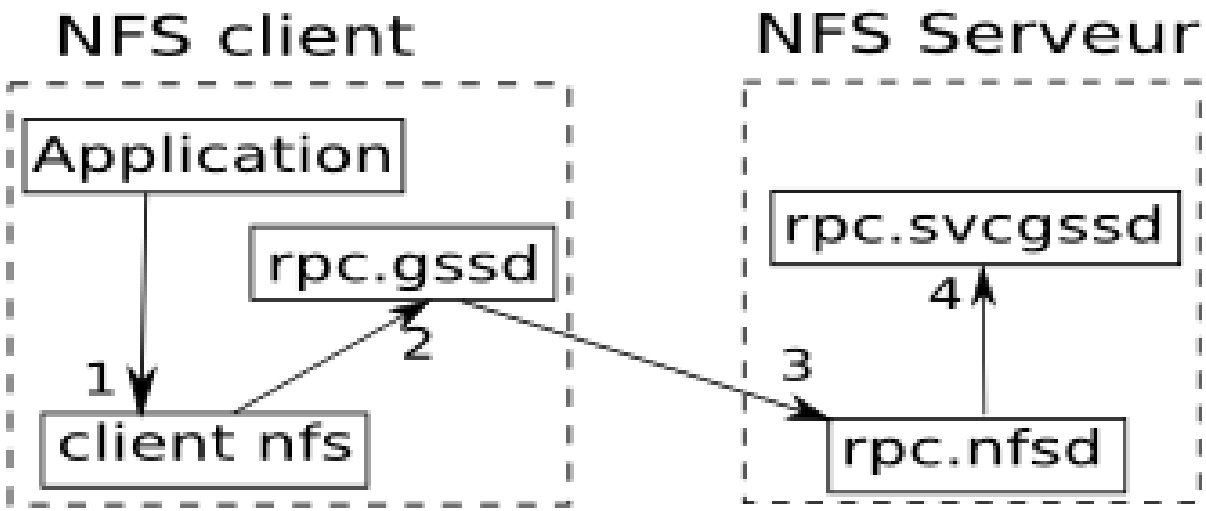
RPCSEC_GSS(1)

1. application : accès objet NFS sur partage en mode krb5
2. client nfs : défaut de cache pour contexte GSS utilisateur/serveur => demande de création à rpc.gssd
3. service gssd : demande un ST nfs/serveur@REALM crée un nouveau contexte GSS avec le ST => rpc.nfsd requête de type NULL
4. service nfsd : transmet ce contexte à rpc.svcgssd



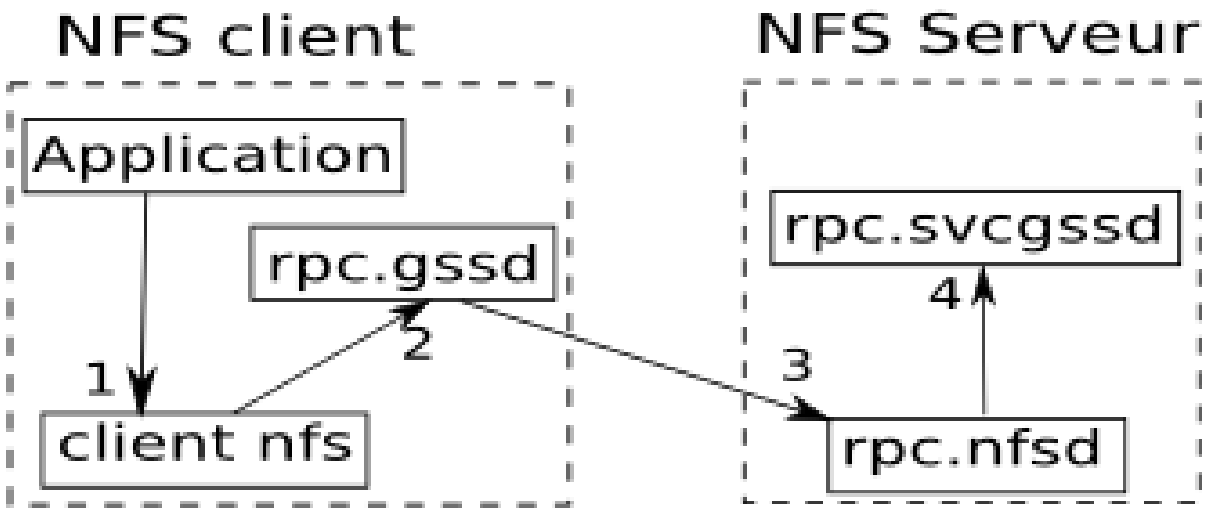
RPCSEC_GSS(1)

1. application : accès objet NFS sur partage en mode krb5
2. client nfs : défaut de cache pour contexte GSS utilisateur/serveur => demande de création à rpc.gssd
3. service gssd : demande un ST nfs/serveur@REALM crée un nouveau contexte GSS avec le ST => rpc.nfsd requête de type NULL
4. service nfsd : transmet ce contexte à rpc.svcgssd



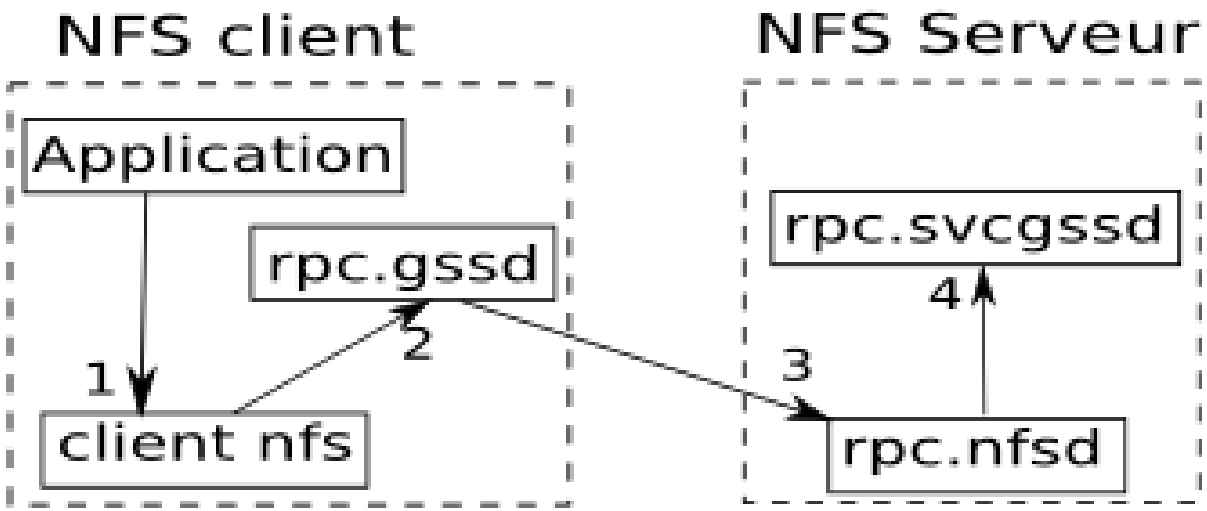
RPCSEC_GSS(1)

1. application : accès objet NFS sur partage en mode krb5
2. client nfs : défaut de cache pour contexte GSS utilisateur/serveur => demande de création à rpc.gssd
3. service gssd : demande un ST nfs/serveur@REALM crée un nouveau contexte GSS avec le ST => rpc.nfsd requête de type NULL
4. service nfsd : transmet ce contexte à rpc.svcgssd



RPCSEC_GSS(1)

1. application : accès objet NFS sur partage en mode krb5
2. client nfs : défaut de cache pour contexte GSS utilisateur/serveur => demande de création à rpc.gssd
3. service gssd : demande un ST nfs/serveur@REALM crée un nouveau contexte GSS avec le ST => rpc.nfsd requête de type NULL
4. service nfsd : transmet ce contexte à rpc.svcgssd

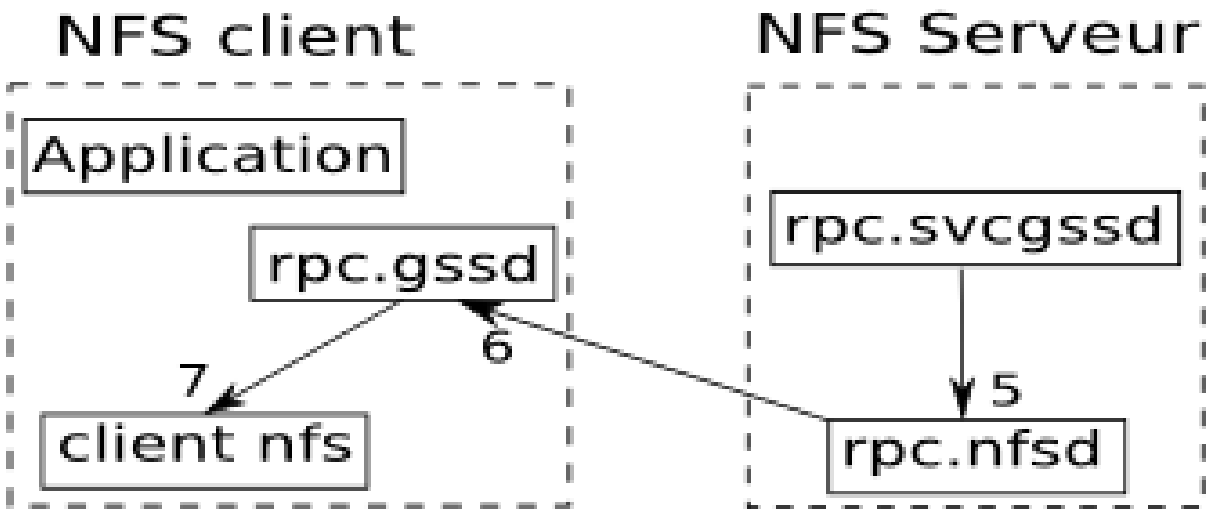


RPCSEC_GSS(2)

5 service svcgssd : vérifie validité du ST ; complète contexte GSS utilisateur/serveur avec un secret pour les échanges NFS ; le transmet à rpc.nfsd

6 service nfsd : cache le contexte GSS et répond à rpc.gssd du client avec le contexte complet

7 rpc.gssd : fournit contexte pour stockage dans noyau

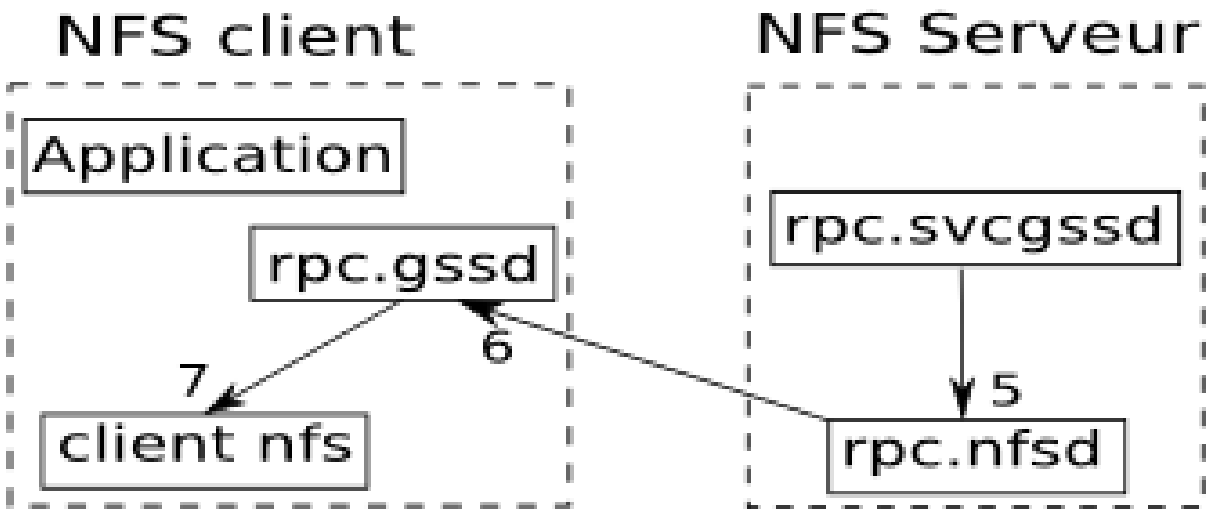


RPCSEC_GSS(2)

5 service svcgssd : vérifie validité du ST ; complète contexte GSS utilisateur/serveur avec un secret pour les échanges NFS ; le transmet à rpc.nfsd

6 service nfsd : cache le contexte GSS et répond à rpc.gssd du client avec le contexte complet

7 rpc.gssd : fournit contexte pour stockage dans noyau

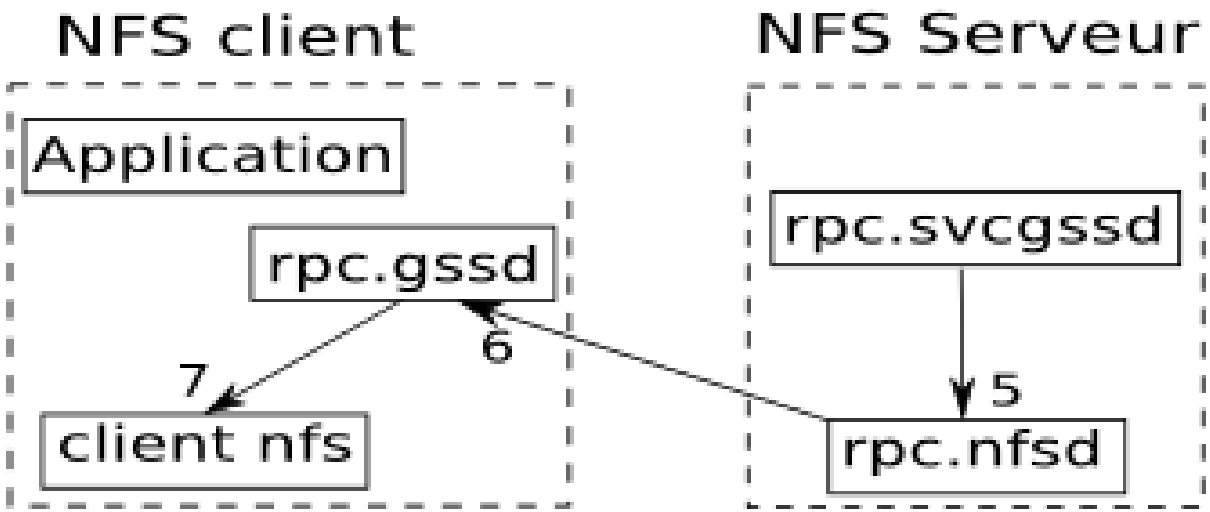


RPCSEC_GSS(2)

5 service svcgssd : vérifie validité du ST ; complète contexte GSS utilisateur/serveur avec un secret pour les échanges NFS ; le transmet à rpc.nfsd

6 service nfsd : cache le contexte GSS et répond à rpc.gssd du client avec le contexte complet

7 rpc.gssd : fournit contexte pour stockage dans noyau

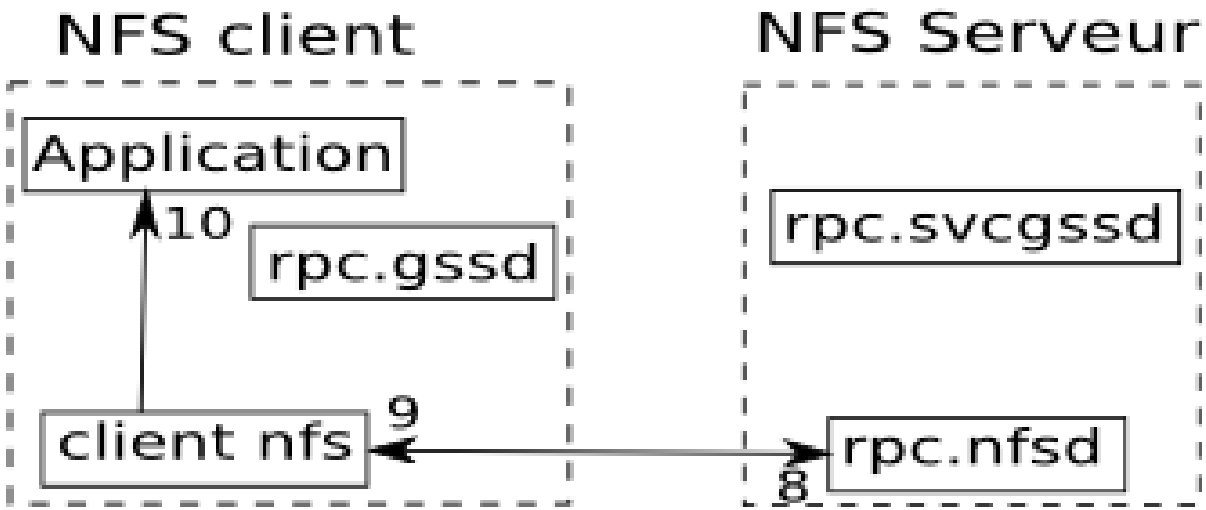


RPCSEC_GSS(3)

8 client nfs : requête NFS avec mode de protection demandée à l'aide du secret dans le contexte GSS

9 nfsd : contexte GSS => réception et renvoi réponse selon protection demandée

10 nfs fournit la réponse à l'application.

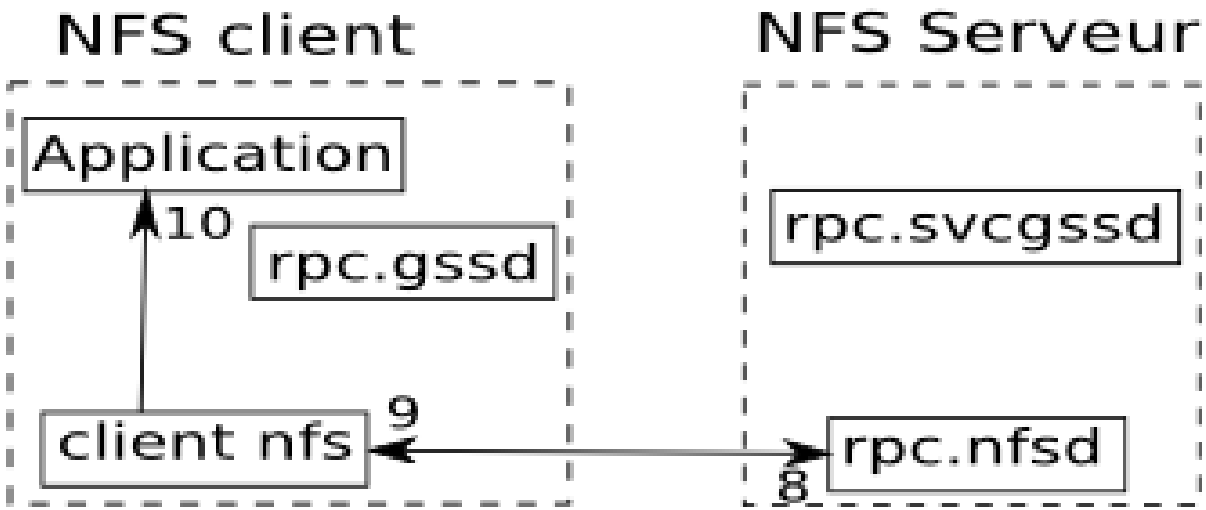


RPCSEC_GSS(3)

8 client nfs : requête NFS avec mode de protection demandée à l'aide du secret dans le contexte GSS

9 nfsd : contexte GSS => réception et renvoi réponse selon protection demandée

10 nfs fournit la réponse à l'application.

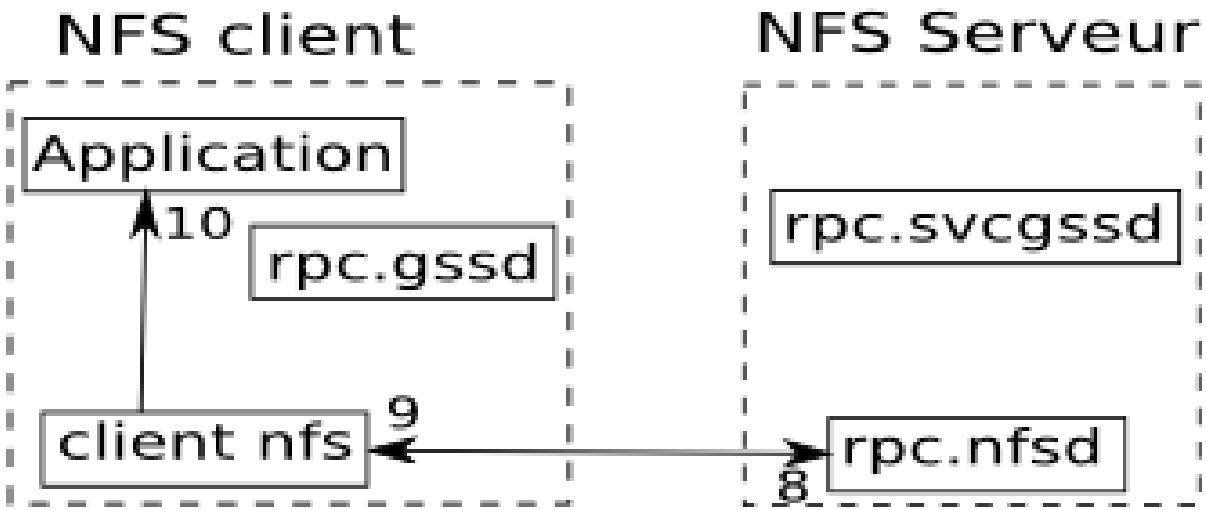


RPCSEC_GSS(3)

8 client nfs : requête NFS avec mode de protection demandée à l'aide du secret dans le contexte GSS

9 nfsd : contexte GSS => réception et renvoi réponse selon protection demandée

10 nfs fournit la réponse à l'application.



Progression

Conclusion

Liste des services RPC

NFSv3		NFSv4	
client	serveur	client	serveur
rpc.lockd rpc.statd	rpc.nfsd rpc.mountd rpc.lockd rpc.statd	rpc.idmapd rpc.gssd	rpc.nfsd rpc.idmapd rpc.svcgssd

La mise en pratique avec le TP après la pause...

Vos questions ?