Secrets d'authentification sous Windows

Aurélien Bordes
aurelien26@free.fr
http://aurelien26.free.fr/secauth/

v1.0 – OSSIR Windows – 8 octobre 2007

Plan

- Rappel
- Authentification locale
  - condensats LM et NTLM
  - sessions d'authentification
  - comptes de services / comptes machine
- Authentification distante
  - protocole LM, NTLM, NTLMv2
- Focus sur SMB
  - base des partages

Authentification

Mécanisme qui consiste à prouver une identité dont se réclame une entité. Généralement, l'authentification est précédée d'une identification qui permet à cette entité de se faire reconnaître du système

Formes d’authentification

- Deux formes d’authentification sont à distinguer :
  - **locale** à un système (*Interactive Authentication*)
  - **à distance** (*Noninteractive Authentication*) via la mise en œuvre d’un protocole d’authentification

Authentification locale (*Interactive Authentication*)

- Permet de valider les demandes d'authentification locale à la machines

- Cadre d'utilisation :
  - authentification des utilisateurs
  - authentification des comptes de services
  - création de la session d'authentification de la machine

Rappels

- Pour valider un compte, le plus simple est de le comparer à un mot de passe stocké localement
- Il est préférable de ne pas stocker les mots de passe en clair, mais une forme non réversible, par exemple un condensat (*hash*)

- Deux formes de condensats cohabitent sous Windows :
  - condensat LM
  - condensat NTLM

Comparaison LM / NTLM

	LM	NTLM
Apparition	Historique	Windows NT4 SP3
Taille	14 caractères (2x7)	255 caractères
Alphabet	Restreint (OEM)	Large (Unicode)
Algorithme de calcul	DES	MD4
Graine	NON	NON

Session d'authentification (*logon session*)

- Pour chaque entité qui s'authentifie localement, une **session d'authentification** est créée. Elle représente un contexte d'authentification associé aux processus ou aux *threads*

- Une session est composée :
  - d'un identifiant unique de session (*logonID*)
  - le domaine et le nom de l'utilisateur
  - l'identifiant de sécurité (SID) de l'utilisateur
  - le nom de l'*Authentication Package* ayant validé et créé la session
  - le type de session (Interactive, Réseau, Service)

Fonctions LSASS

- Les sessions d'authentification sont gérées par LSASS à l'aide des fonctions (LSASS) :
  - **`CreateLogonSession`**
  - **`DeleteLogonSession`**
- Les fonctions suivantes permettent d'obtenir des informations sur les sessions en cours (secur32.dll) :
  - **`LsaEnumerateLogonSessions`**
  - **`LsaGetLogonSessionData`**

Association de *credentials*

- Il est possible d'associer des *credentials* aux sessions d'authentification

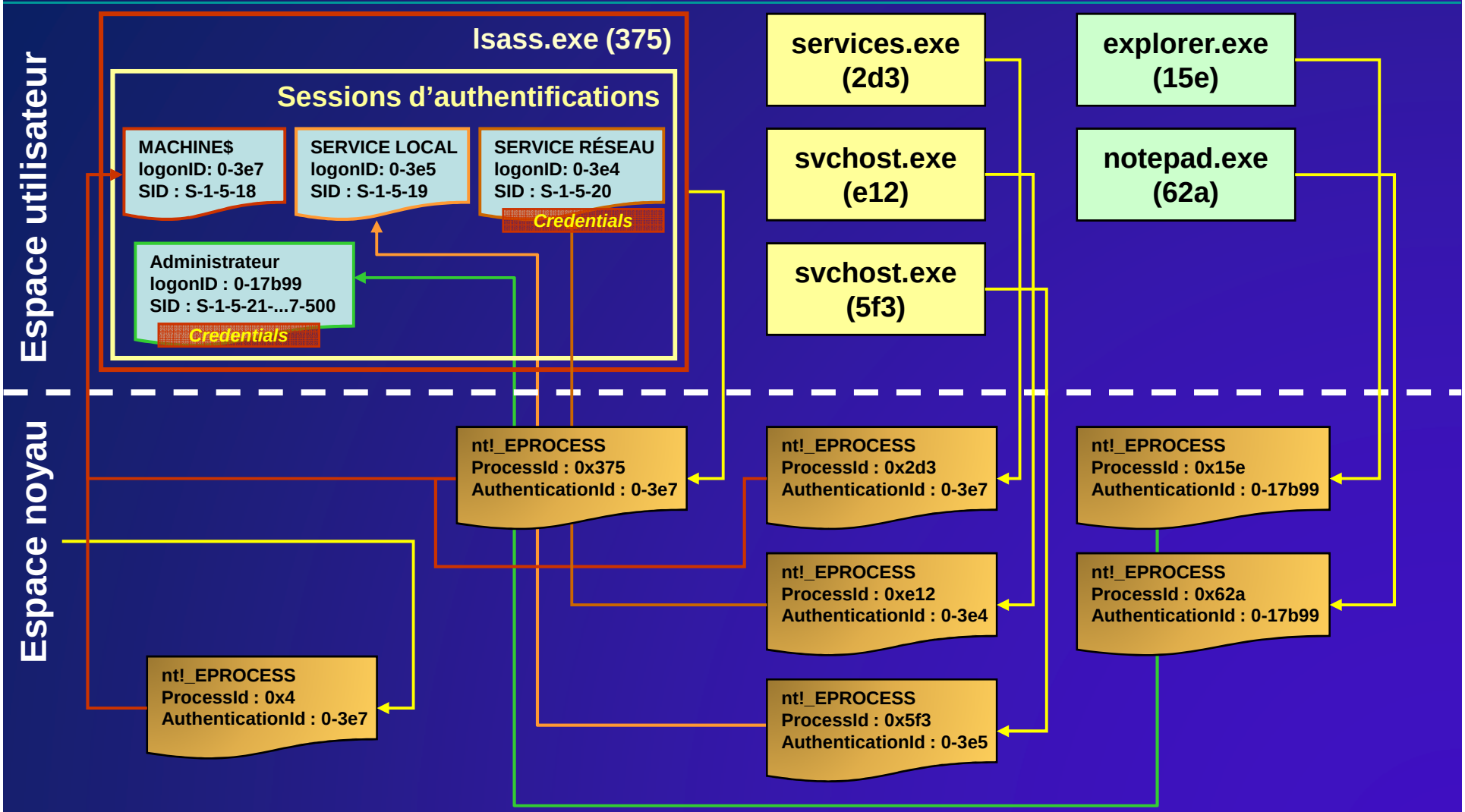
- Les fonctions suivantes permettent de gérer l'association des *credentials* aux sessions (LSASS) :
  - **`AddCredential`**
  - **`GetCredentials`**
  - **`DeleteCredential`**

Association
jeton de sécurité / session d'authentification

- Le noyau gère les jetons de sécurité (*access token*)
- Le champ **`nt!_EPROCESS.Token.AuthenticationId`** indique le *logonID* de la session d'authentification associée

- Un jeton de sécurité est toujours associé à un processus
- Un jeton de sécurité peut être associé à un thread dans le cas d'un changement de contexte (*impersonation*)

Association ProcessID / LogonID



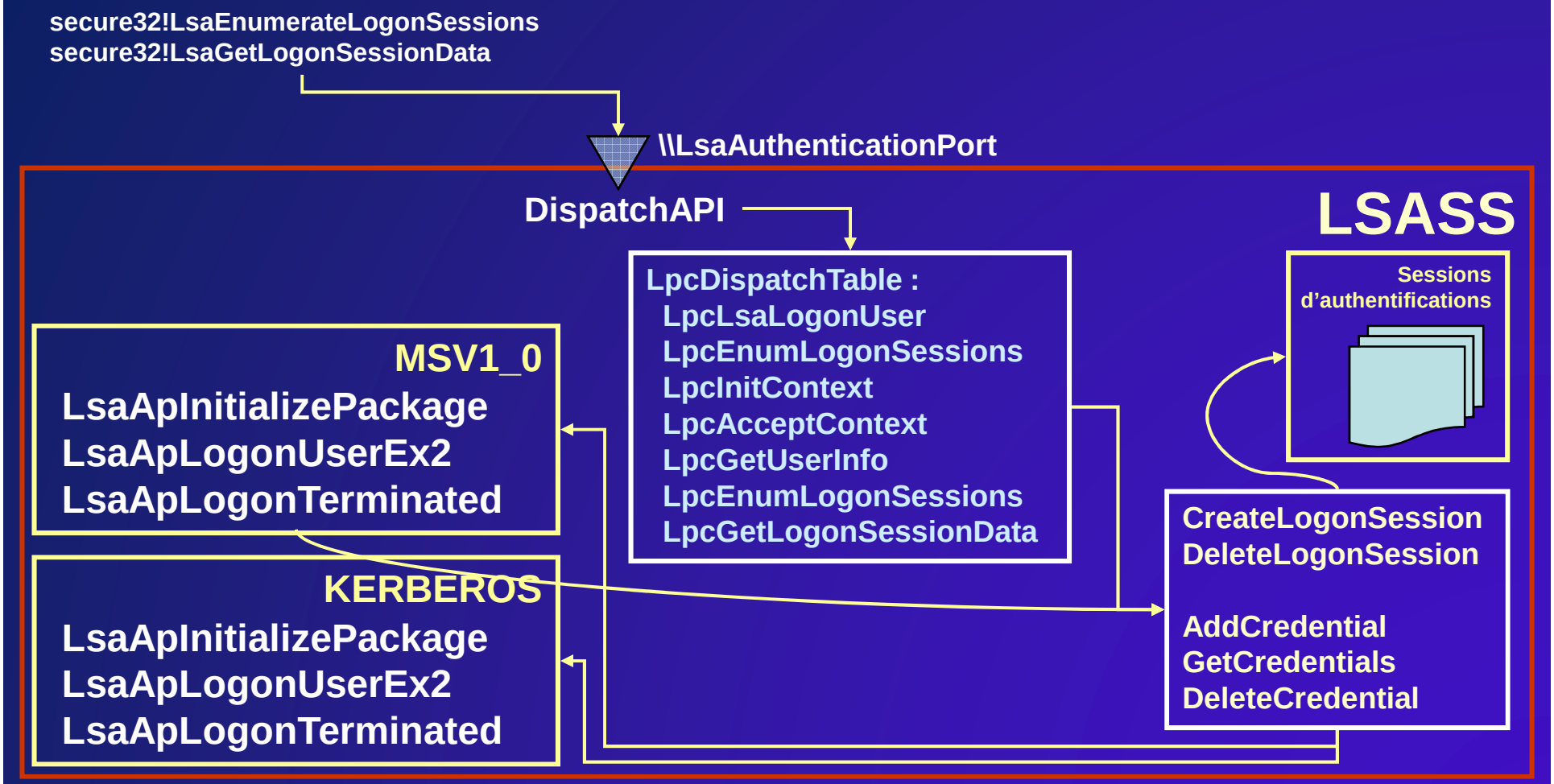
Authentication Packages

- LSASS appelle des librairies (DLL) dénommées « *Authentication Packages* » pour valider les authentifications locales
- Ces librairies sont chargées dans le processus LSASS
- Un AP doit implémenter les fonctions appelées par LSASS, en particulier :
  - **`LsaApInitializePackage`**
  - **`LsaApLogonUserEx2`**
  - **`LsaApLogonTerminated`**

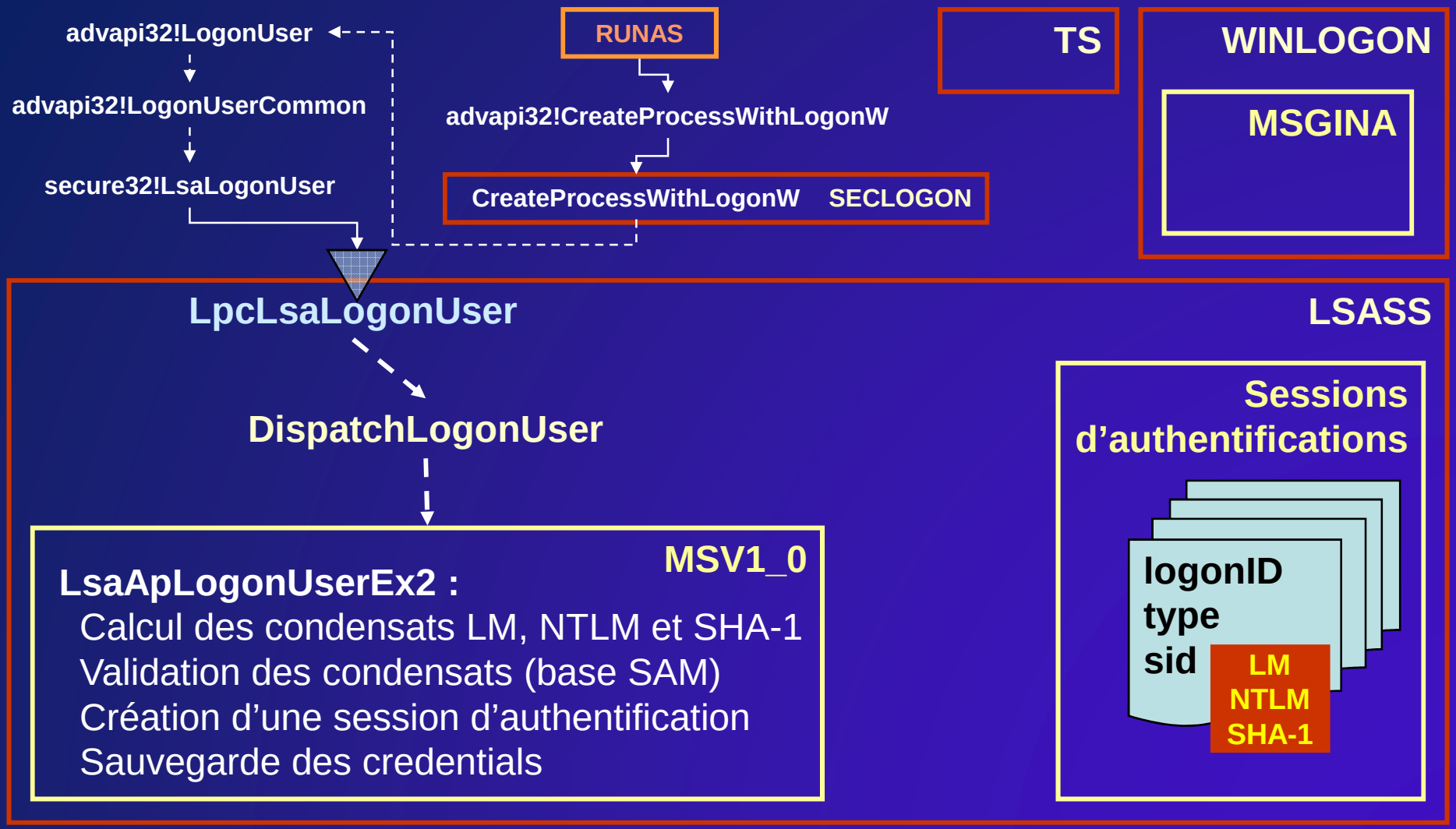
AP fournis en standard

- Windows dispose en standard des AP :
  - **msv1_0**
    - authentification locale (base SAM)
    - authentification domaine pré-Windows 2000
  - **Kerberos**
    - authentification dans un domaine Active Directory

Schéma d'appels à LSASS



Processus d'une authentification locale de type interactive (LogonType=2)

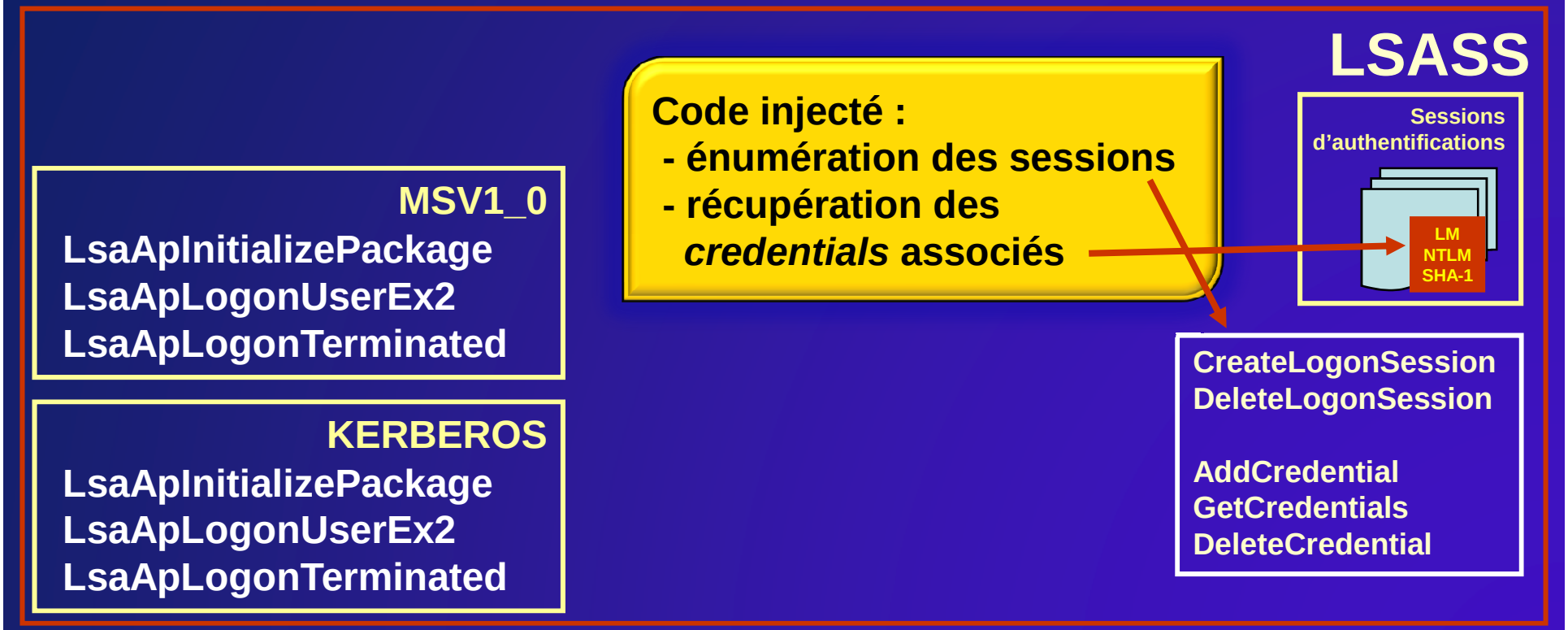


Démo 1 : récupération des condensats LM

- Même si le condensat LM n'est pas stocké localement (désactivé via **`NoLmHash`** ou utilisation d'un compte du domaine), il est stocké en mémoire dans la session d'authentification

- Les sessions d'authentification ne sont pas immédiatement supprimées

Démo 1 : récupération des condensats LM



Comptes de services

Nom	Identité	SID	logonID	Privilèges	Authentification réseau
LocalSystem	SYSTEM	S-1-5-18	0-3e7	Forts	
NetworkService	SERVICE RÉSEAU	S-1-5-20	0-3e4	Faibles	Compte machine [1]
LocalService	SERVICE LOCAL	S-1-5-19	0-3e5	Faibles	Anonyme
Compte explicite	Cf. compte	Cf. compte	Variable	Cf. compte	Cf. compte

[1] Pour une machine membre d'un domaine
(mot de passe vide pour une machine autonome)

Processus d'une authentification locale de type service (LogonType=5)

- Processus identique à celui de type interactive sauf que le mot de passe est récupéré dans les secrets de la LSA
- L'entrée correspond au nom du service, préfixé par **_SC_**

Authentification dans le contexte « machine »

- Si un système est membre d'un domaine, la machine dispose d'un compte dans l'annuaire (**`machine$`**)
- Le mot de passe associé est stocké dans les secrets de la LSA sous l'entrée **`$MACHINE.ACC`**
- Ces authentifiants sont associés à la session du SERVICE RÉSEAU (condensat NTLM)

- Il est possible de s'authentifier auprès de tout système membre du domaine avec un compte d'une machine

Démo 2 : Comptes de service et compte machine

- Récupération des secrets de la LSA
- Analyse les entrées dont le préfixe est **_SC_**
- Récupération de l'entrée **$MACHINE.ACC**
- Connexion via ce mot de passe

Authentification distante (*Noninteractive Authentication*)

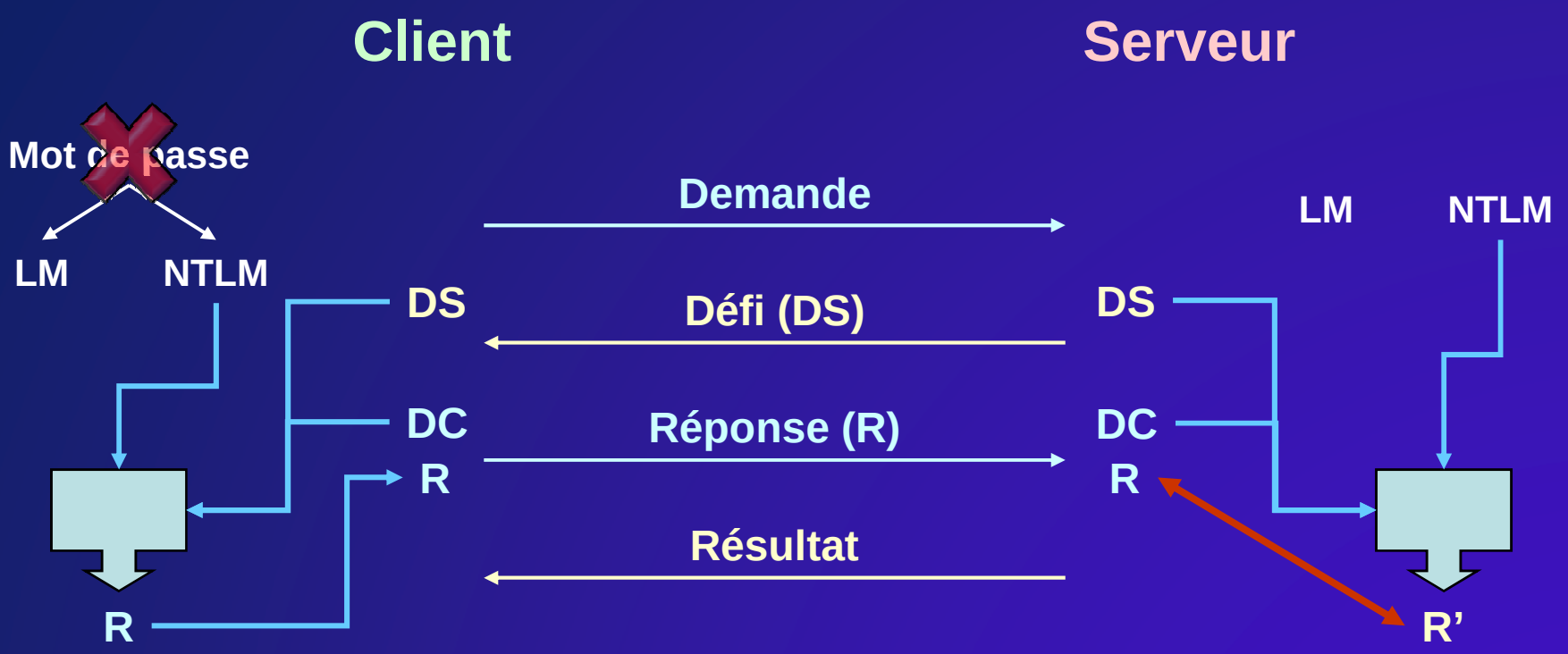
- Permet de s'authentifier à distance via l'utilisation d'un protocole d'authentification
- Plusieurs formes possibles :
  - clair
  - défi / réponse
  - formes avancées (dérivation, …)
- Généralement, on doit utiliser un protocole qui permette de ne pas dévoiler ses authentifiants

Protocoles d'authentification LM/NTLM

- Sous Windows, plusieurs protocoles sont disponibles :
  - Protocole historique, basé sur le condensat LM :
    - LM
  - Nouveaux protocoles, basés sur le condensat NTLM :
    - NTLM
    - NTLM2
    - NTLMv2

- La clé de registre **`LmCompatibilityLevel`** permet d'activer ou de refuser ces protocoles

Schéma général du défi / réponse



DS : défi serveur **R : réponse client**

DC : défi client **R' : réponse calculée**

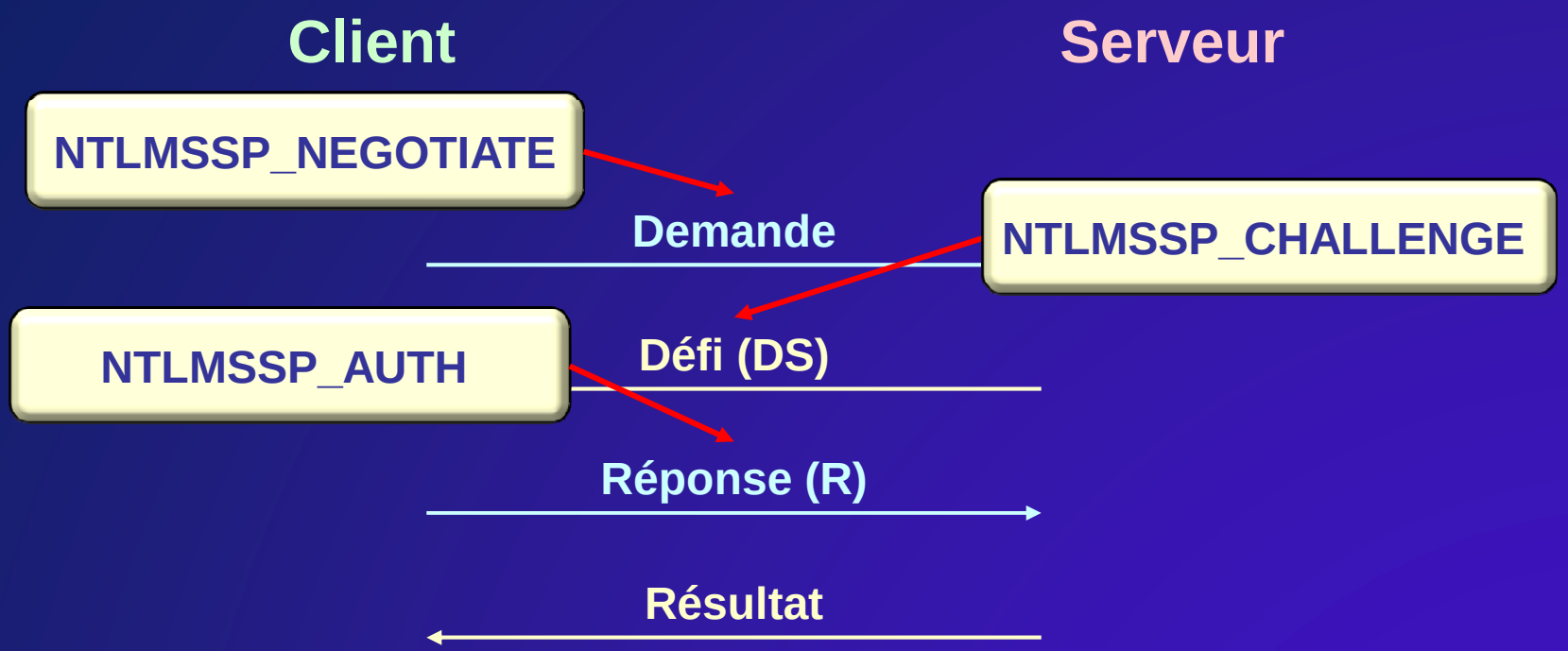
Comparaison protocoles

Protocole	Condensat utilisé	Défi serveur	Défi client	Calcul réponse
LM	LM	OUI	NON	DES
NTLM	NTLM	OUI	NON	DES
NTLM2	NTLM	OUI	OUI	DES
NTLMv2	NTLM	OUI	OUI	HMAC-MD5

NTLMSSP

- Le protocole NTLMSSP permet de structurer les échanges des protocoles LM, NTLM, …
- Le protocole définit trois types de message :
  - **`NTLMSSP_NEGOTIATE`** (Client → Serveur) : démarre une négociation
  - **`NTLMSSP_CHALLENGE`** (Serveur → Client) : permet au serveur d'envoyer son défi
  - **`NTLMSSP_AUTH`** (Client → Serveur) : permet au client d'envoyer ses réponses (champs LM et NTLM)

NTLMSSP


Client
Serveur
NTLMSSP_NEGOTIATE
Demande
NTLMSSP_CHALLENGE
NTLMSSP_AUTH
Défi (DS)
Réponse (R)
Résultat

Security Support Provider (SSP)

- Les protocoles d'authentification sont implémentés par les librairies dénommées **Security Support Provider**
- Principaux SSP fournis en standard :
  - **msv1_0** : implémente NTLMSSP
  - **Kerberos** : implémente Kerberos v5
  - **Negociate** : négocie l'utilisation des deux précédents :
    - Kerberos pour les membres d'un domaine
    - NTLMSSP dans les autres cas

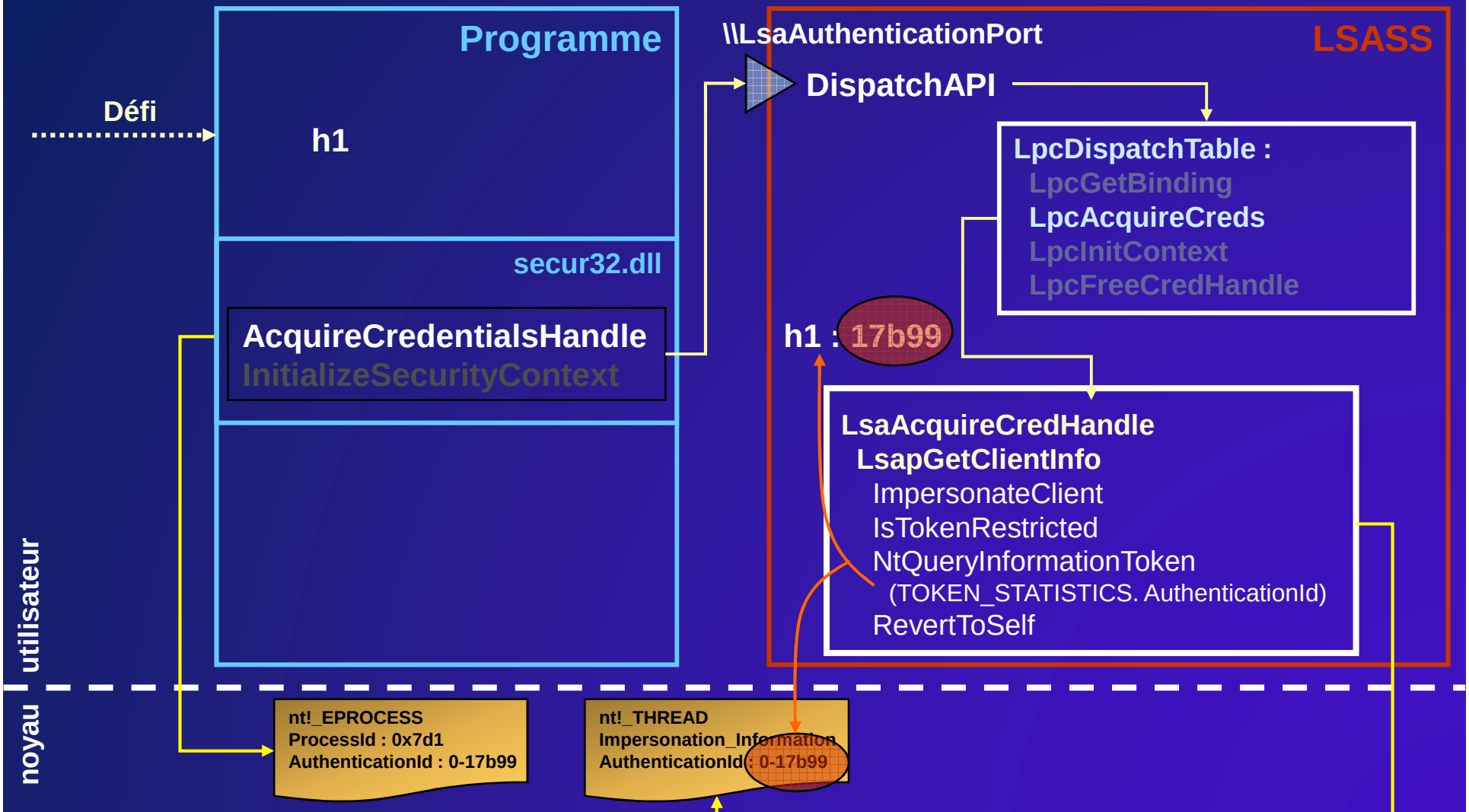
Security Support Provider Interface (SSPI)

- L'API SSPI met en œuvre l'authentification via les SSP
- Les fonctions principales sont (secur32.dll) :
  - **AcquireCredentialsHandle** : initialise un SSP et permet de spécifier le contexte d'authentification
  - **InitializeSecurityContext** : permet au client de générer les messages d'authentification à envoyer au serveur
  - **AcceptSecurityContext** : permet au serveur de valider les messages reçus du client

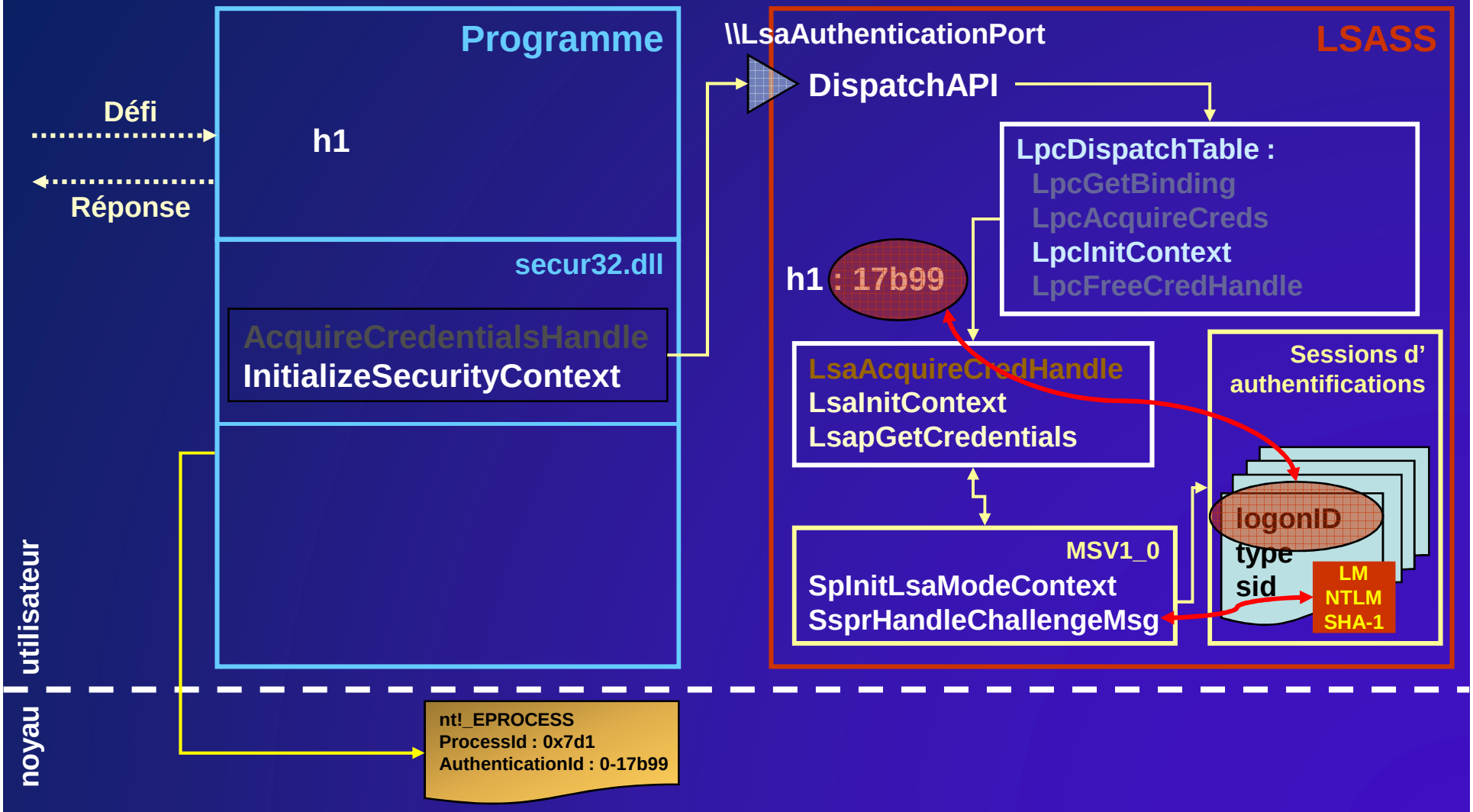
Contexte d'authentification

- Le client spécifie son contexte d'authentification via le paramètre **`pAuthData`** lors de l'appel à la fonction **`AcquireCredentialsHandle`** :
  - **`pAuthData`** = structure spécifiant {User, Domain, Password}
  - **`pAuthData`** = NULL (*default context*) : ce sont les *credentials* de l'utilisateur (ceux de sa session) qui seront utilisés
- Dans le deuxième cas, le package doit accéder aux *credentials* associés à la session de l'utilisateur

Fonctionnement de AcquireCredentialsHandle



Fonctionnement de InitializeSecurityContext



Authentification avec NTLMSSP

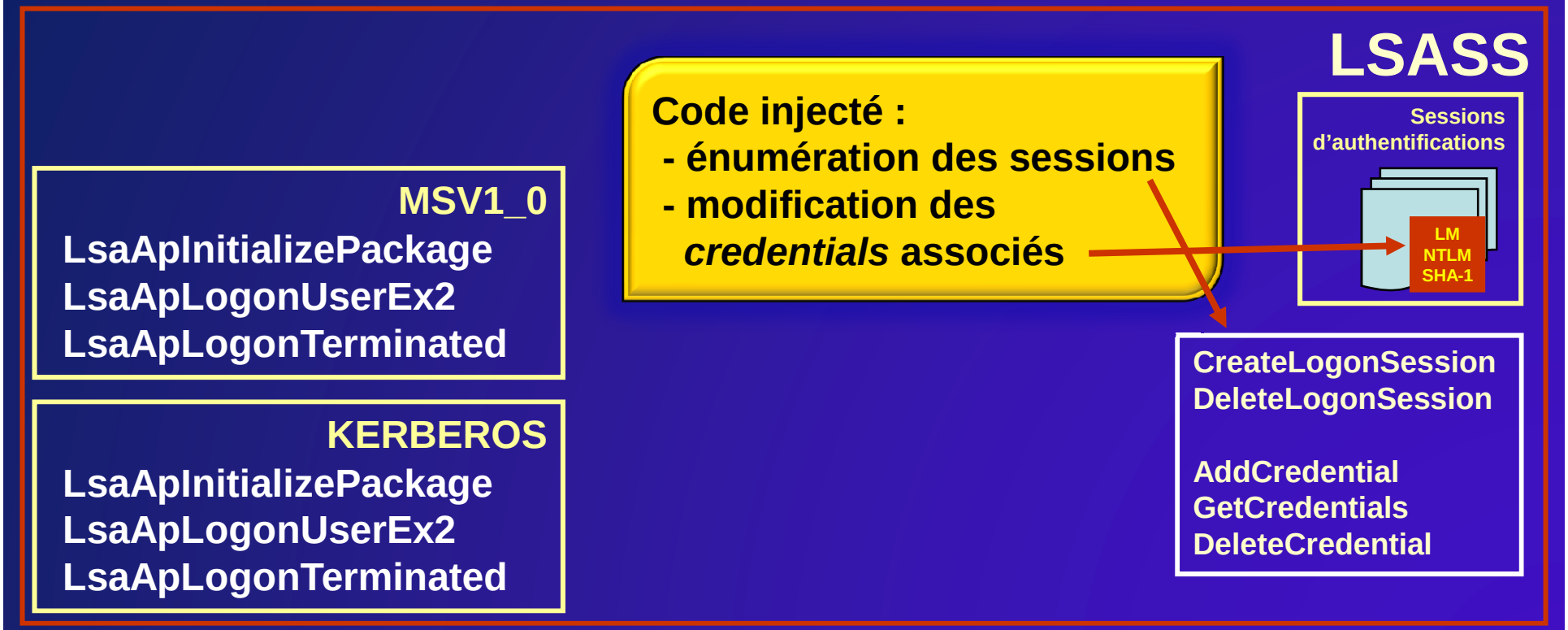
- Le mot de passe n'est pas nécessaire pour répondre aux défis : il suffit de connaître uniquement le condensat
- Il est ainsi possible de s'authentifier à distance uniquement en connaissant le condensat LM ou NTLM

Démo 3 : changement de *credentials*

- Changement des *credentials* associés à une session d'authentification

- Techniques :
  - récupérer le condensat LM ou NTLM d'un compte
  - éditer les *credentials* d'une session
  - se connecter sur une machine à distance avec le nouveau contexte

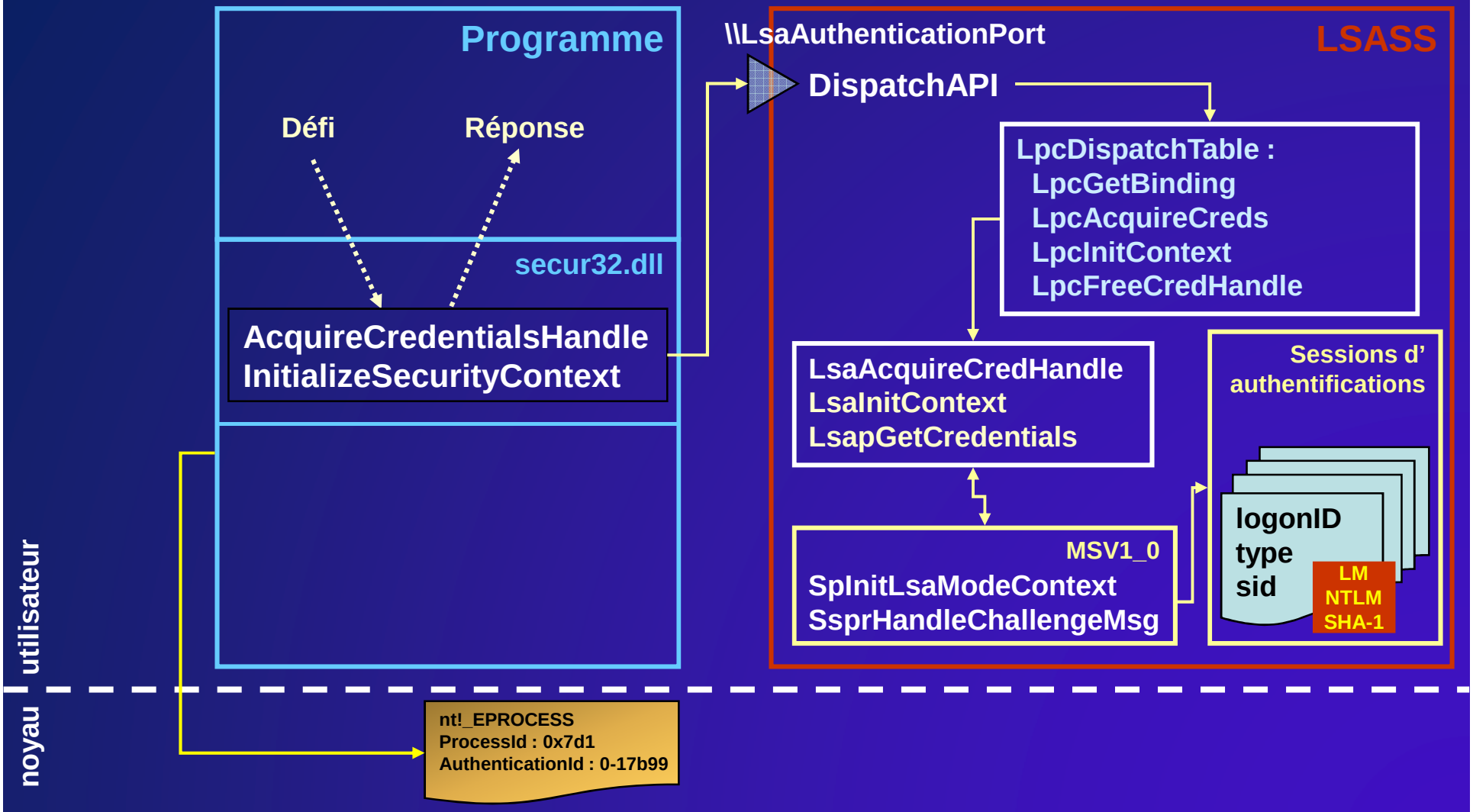
Démo 3 : changement de *credentials*



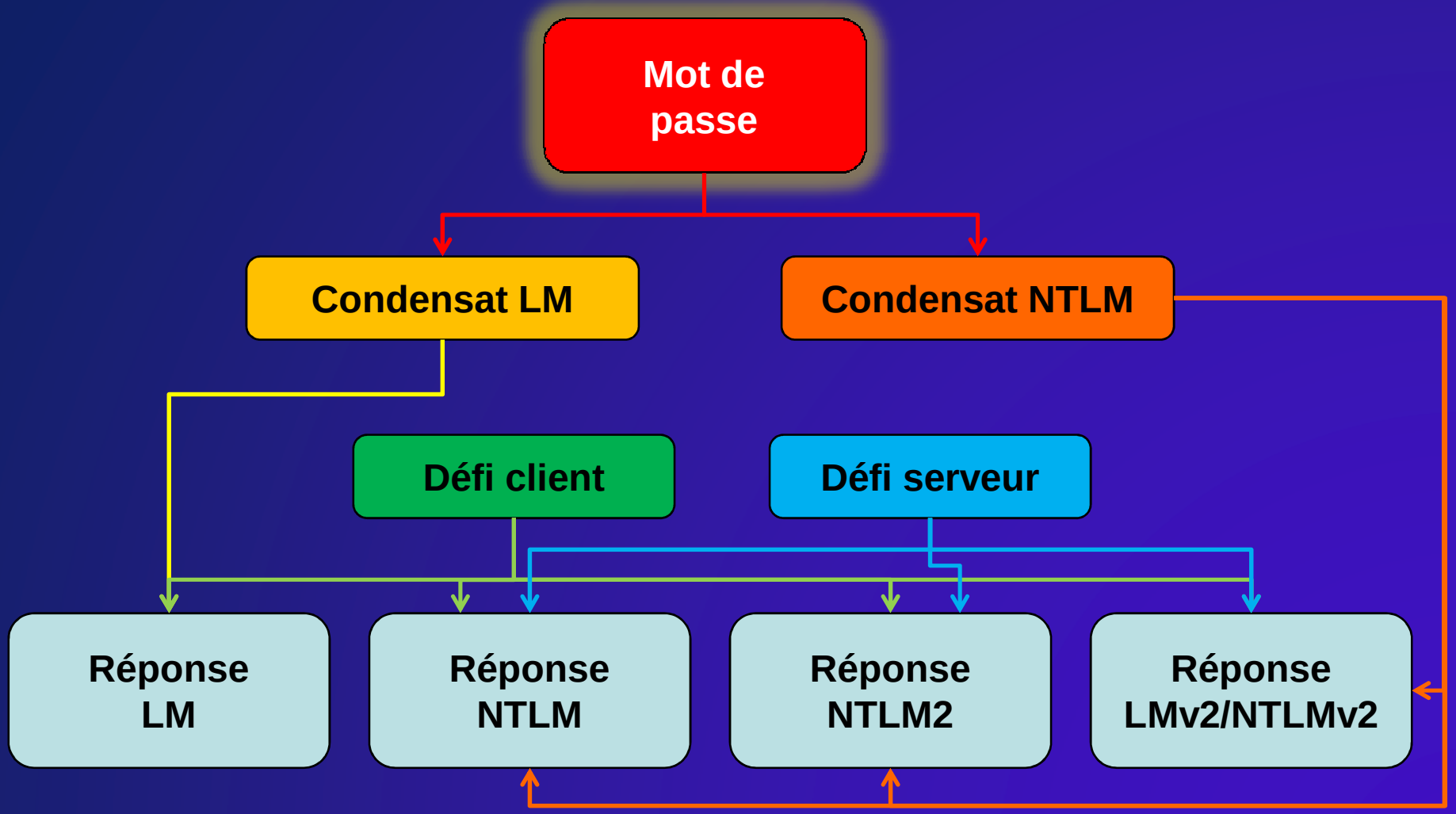
Démo 4 : récupération d'un défi / réponse

- Un appel aux fonctions **`AcquireCredentialsHandle`** et **`InitializeSecurityContext`** permet de récupérer un défi/réponse LM/NTLM
- Il est possible à partir de cette capture de réaliser différentes attaques pour tenter de retrouver le mot de passe de l'utilisateur
- Techniques :
  - paramétrer SSPI pour utiliser le SSP NTLMSSP
  - forger un défi (une fois) au format NTLMSSP
  - essayer l'utilisation de LM puis NTLM (NTLMv2 dans tous les cas)

Démo 4 : récupération d'un défi / réponse



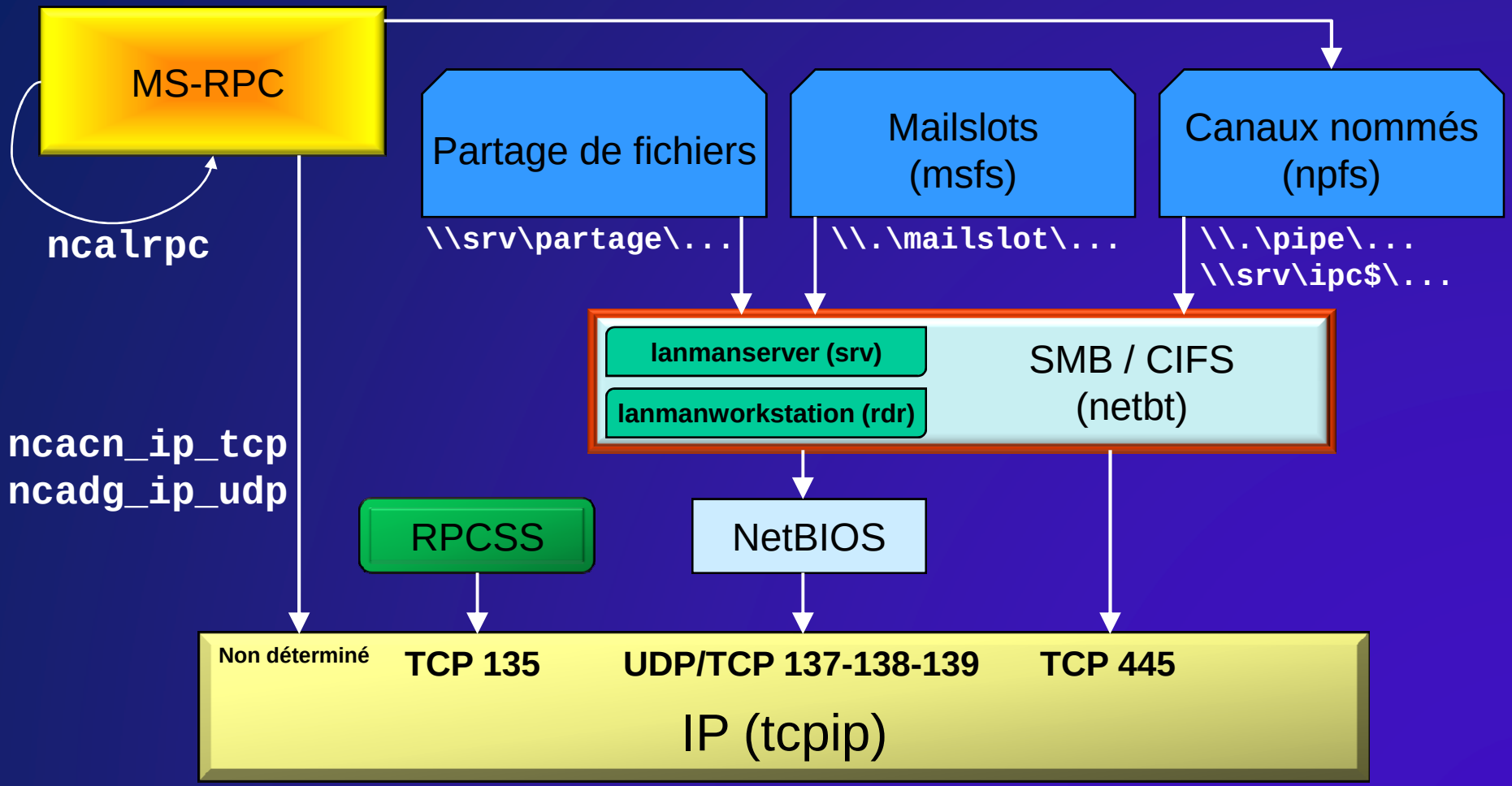
Démo 4 : relations



Application au protocole SMB/CIFS

- Le protocole SMB est un protocole réseau permettant le partage de fichiers (au sens large)
- Il est très utilisé sous Windows

Positionnement de SMB

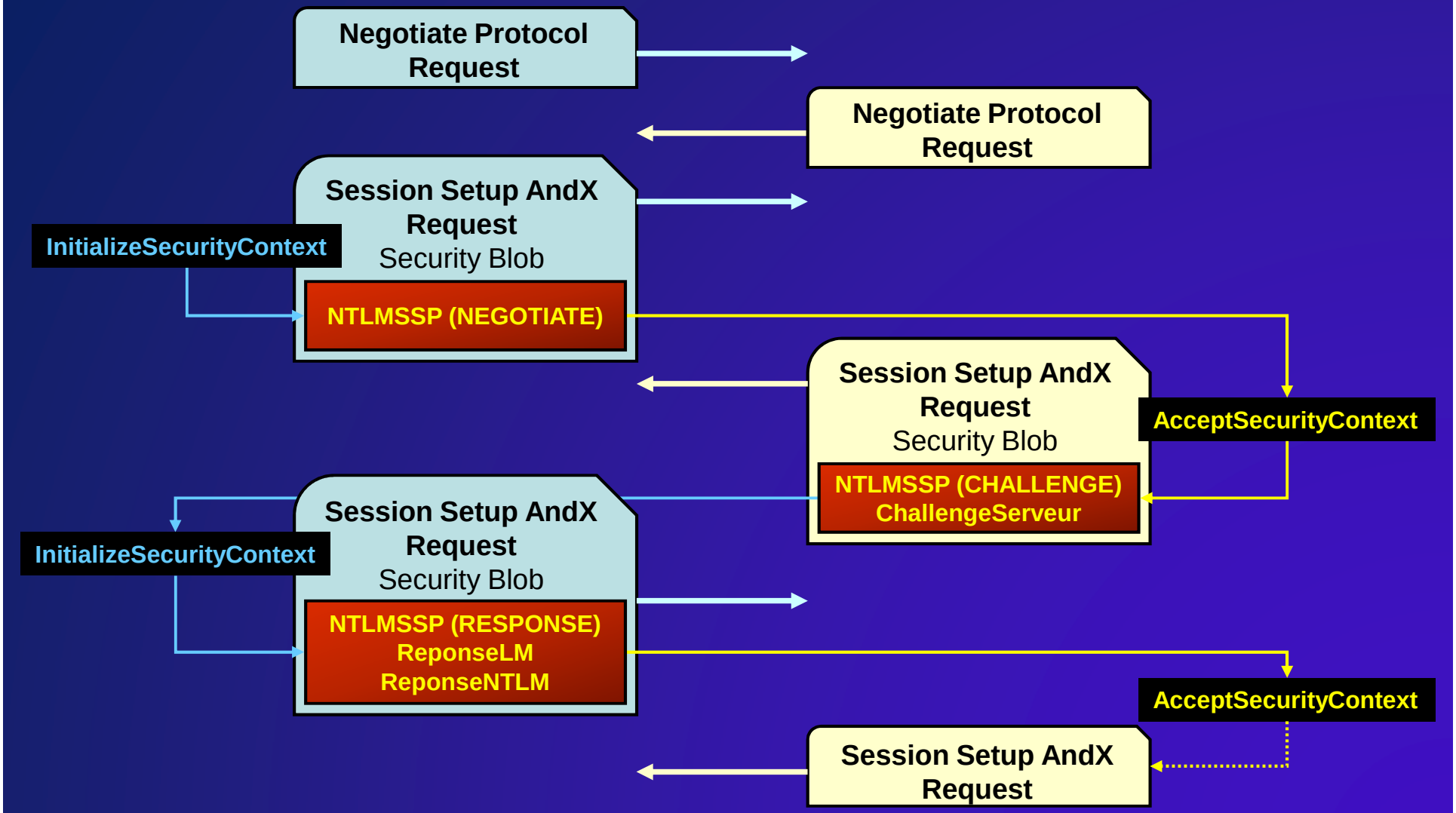


Authentification avec SMB

- SMB repose entièrement sur SSPI et le SSP Negociate pour l'authentification

- Deux contextes pour l'authentification :
  - Utilisateur courant (comportement par défaut pour offrir le SSO)
  - Utilisateur explicite (nécessaire de spécifier un login/mot de passe)

Échanges SMB



Cas de l’utilisateur explicite

- Le nom de l’utilisateur et le mot de passe sont spécifiés au montage du partage
- Cependant, le système est toujours capable de s’authentifier tant que le partage n’est pas supprimé

Base des partages

- Dans l’espace noyau, le pilote mrxsmb.sys gère la base des partages réseaux montés sur le système
- Pour chaque entrée de la base, on trouve :
  - le nom de la machine distante
  - le nombre de partages actifs vers cette machine
  - le numéro de la session d’authentification associée
  - (optionnel) compte alternatif pour l’authentification
    - domaine et nom de l'utilisateur
    - mot de passe en clair

Démo 5 : base SMB

- Visualisation de la base des sessions

Conclusion