SYNACKTIV
DIGITAL SECURITY

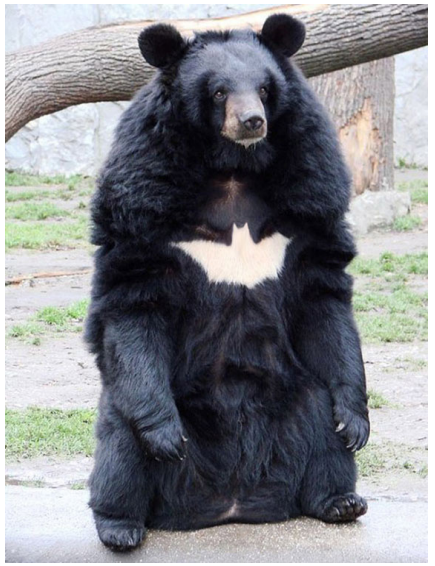
DPAPI exploitation during pentest

Présenté 07/04/2017

Pour STHACK 2017 – rump session

Par Jean-Christophe Delaunay

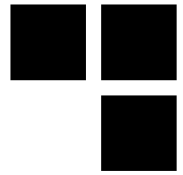
whoami /groups



- **Jean-Christophe Delaunay – @Fist0urs**
- **Jiss/Fist0urs on IRC**
- **Synacktiv – www.synacktiv.ninja**

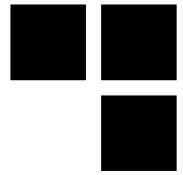
- *Microsoft Windows Active Directory* (kerberom)
- Passcracking – User and contributor to *John The Ripper* and *hashcat* (krb5tgs, axcrypt, keepass, etc.)

What is DPAPI – a bit of history



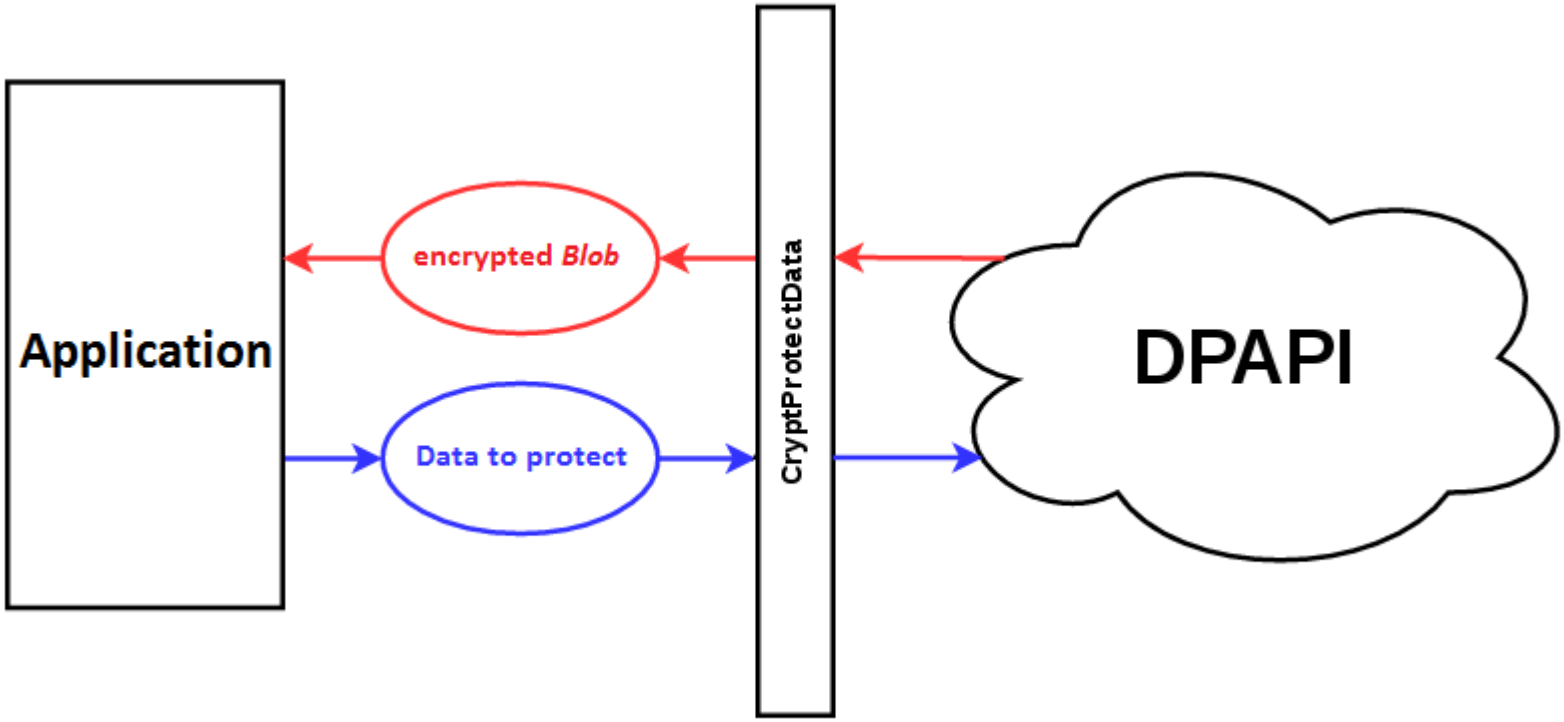
- Data Protection Application Programming Interface
- Helps protect secrets (passwords, certificates, etc.)
- Exists since *Windows 2000*!
- Evolved a lot but core is globally the same
- Invisible for the end-users

What is DPAPI – wtfbbq?

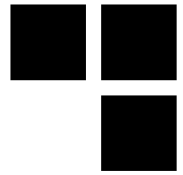


- Cryptography based on user's password (not exactly in fact)
- Easy to implement for developpers:
  - *CryptProtectData*
  - *CryptUnprotectData*
- Widely used:
  - Credential Manager, Windows Vault, IE, Wifi, Certificates, VPN, etc.
  - Google Chrome, GTalk, Skype, Dropbox, iCloud, Safari, etc.

DPAPI Internals – developpers view

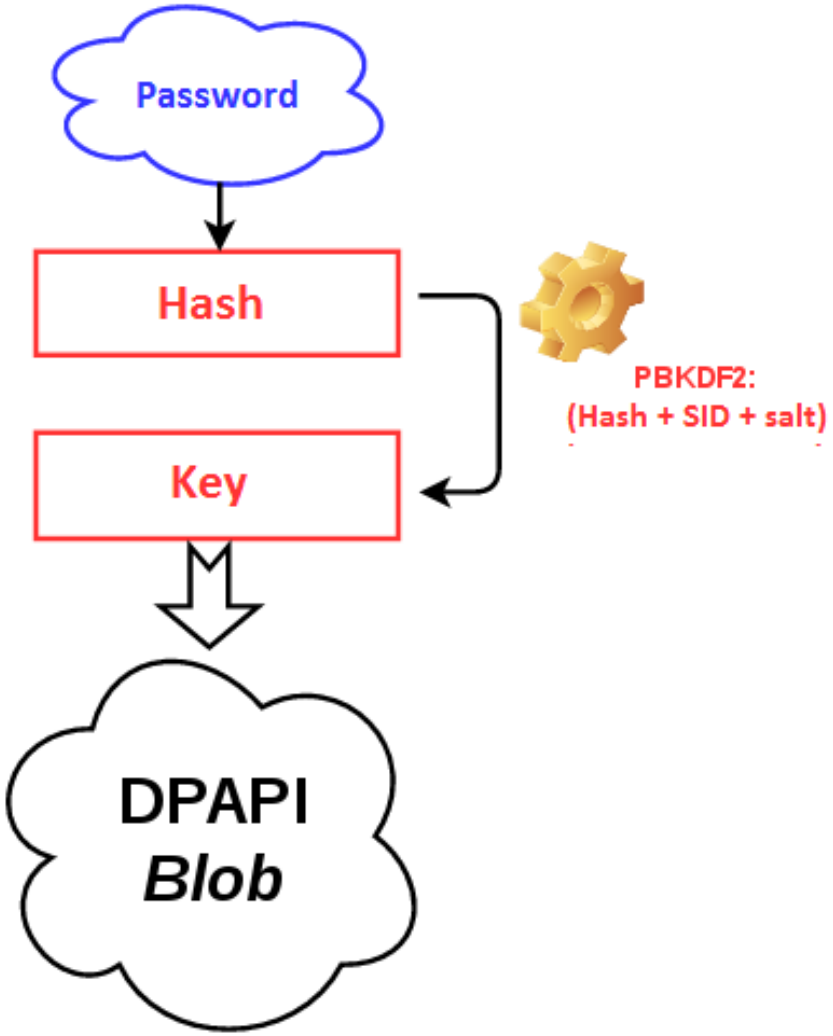


DPAPI Internals – crypto

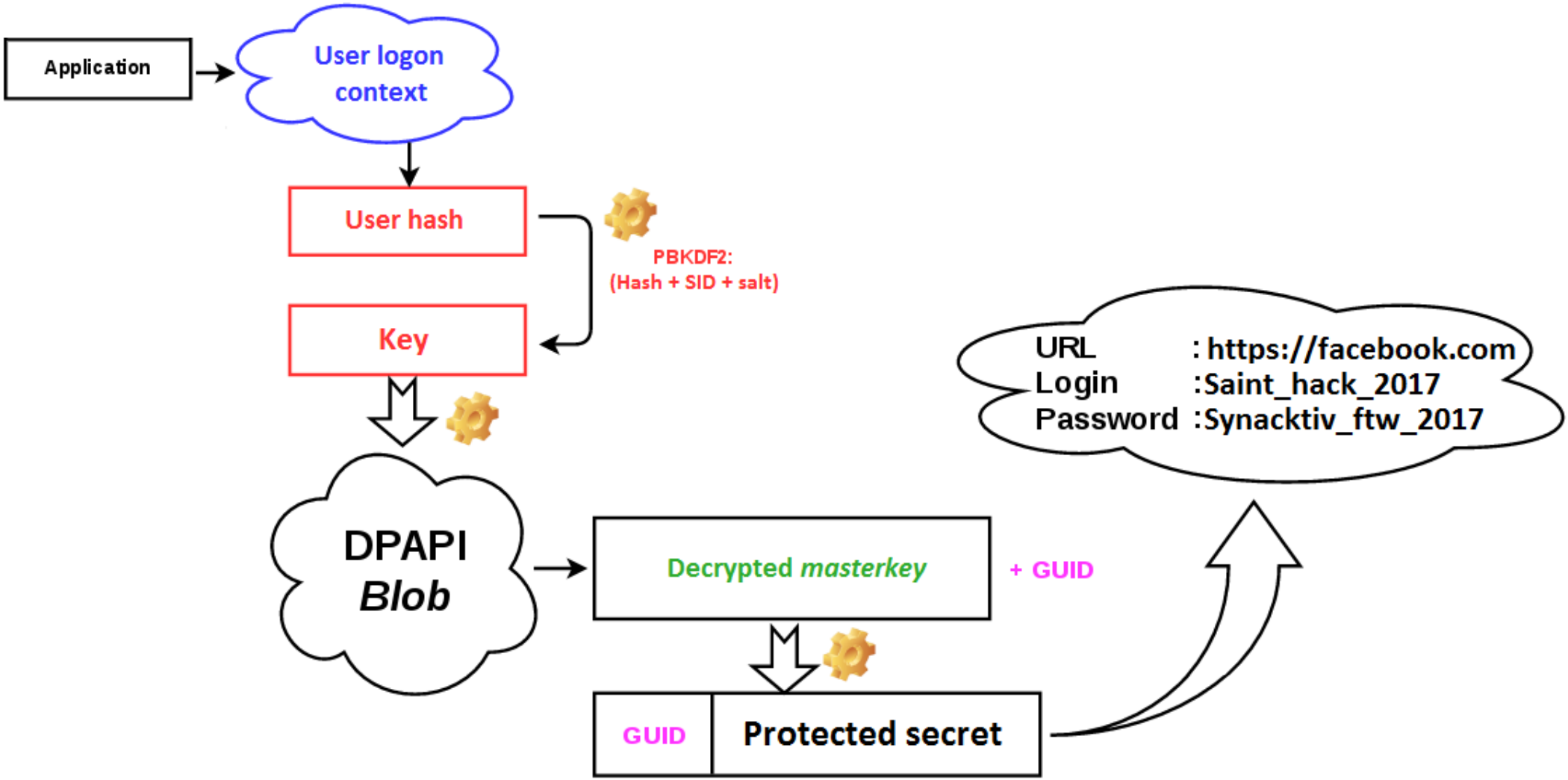


- Secret based on user's password…
- … but this is not secure enough, let's use *master keys*, stored in undocumented *blobs* structures

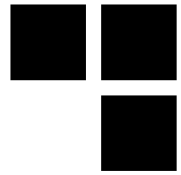
DPAPI Internals – crypto



DPAPI Internals – *overview*



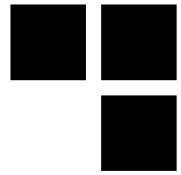
DPAPI Internals – *masterkeys* stored… ?



In the user's profile (%APPDATA %/Roaming/Microsoft)

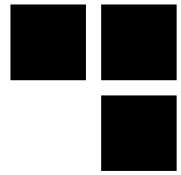
- Protect/SID
  - GUID1
  - GUID2
  - …
  - Preferred

DPAPI – pentests



- 2 possibilities:
  - I can execute some code on the remote host
  - I can’t...

DPAPI – existing tools



- Passcape: shareware + *Windows* only [1]
- impacket: does not decrypt DPAPI protected secrets directly [2]
- mimikatz: extracts secrets *online* and *offline* but *Windows* only [3]
- dpapick: extracts secrets *offline*! First tool published to manage DPAPI *offline*, incredible work! [4]
- dpapilab: an extension of dpapick [5]

DPAPI – pentests

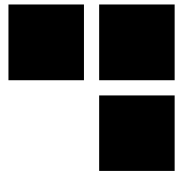
- But wait, you told us that secrets are protected by user's password?...
- ...and *master keys* are also protected by user's password?
- …
- Profit!

DPAPI – pentests

```
Fist0urs@jordy:~/sthack$ python DPAPImk2john.py S-1-5-21-XXXXXXXXX-XXXXXXXXXX-XXXXXXXXXX-1001 2dbd2e3b-XXXX-XXXX-XXXX-519c78c48397
```

```
$DPAPImk$*2*local*S-1-5-21-XXXXXXXXX-XXXXXXXXXX-XXXXXXXXXX-1001*aes256*sha512*8000*1d52563XXXXXXXXXXXXXXXXXa0665d79*288*0049e65595bbXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX7e3b70539567d80afea5168d31c6ccd48b07b8328eb969295611c850f8cf25f06e7f9aede0f5fb4e
```

DPAPI – useful?

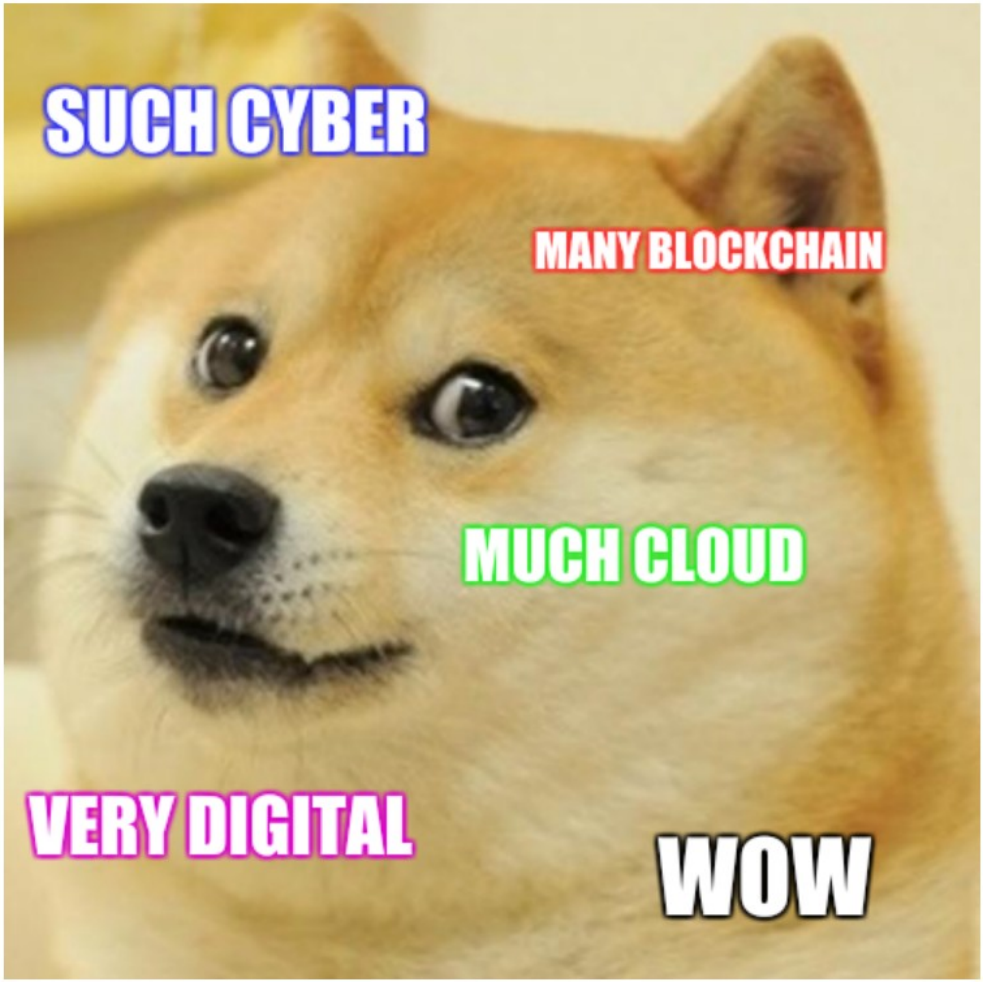


- Created in the roaming profile in an *Active Directory* environnment
- Alternative to *MSCashvX* if computer is hardened (no or only one cached logon hash)
- No need to inject in memory, all you need is a *masterkey* file from the *filesystem* and the user's SID: much more reliable
- Hard to detect compared to existing attacks…
- Difficult to prevent this kind of attack :-/

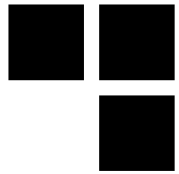
DPAPI – *roadmap*

- Finish the implementation within *John The Ripper*
- Add the implementation within *hashcat*
- Some more things I keep for myself for the moment ;-)

ANY QUESTIONS?



Bibliography



- [1] https://www.passcape.com/
- [2] https://github.com/CoreSecurity/impacket
- [3] http://blog.gentilkiwi.com/mimikatz
- [4] http://dpapick.com/
- [5] https://github.com/dfirfpi/dpapilab