SYNACKTIV

Say Hello to your new cache flow

WHFB and Entra ID

Troopers 2024

26/06/2024

Who are we

SYNACKTIV

- Rémi Jullian
- @netsecurity1
- Reverse engineering

- Geoffrey Bertoli
- @yofbalibump
- Pentest

- Théo Gordyjan
- @___t0___
- Pentest

Who are we

SYNACKTIV

- **Synacktiv**
  - Offensive security company based in France
  - 170 Experts
  - Pentest / Red Team - Reverse Engineering / Vulnerability research – Development - Incident Response
  - Hexacon in Paris (October)

Agenda

SYNACKTIV

- Introduction
- WHFB and Microsoft Entra ID
- Cached data format
- Cached data for offline authentication
- Demo
- A word on DPAPI
- Conclusion

Introduction

Introduction

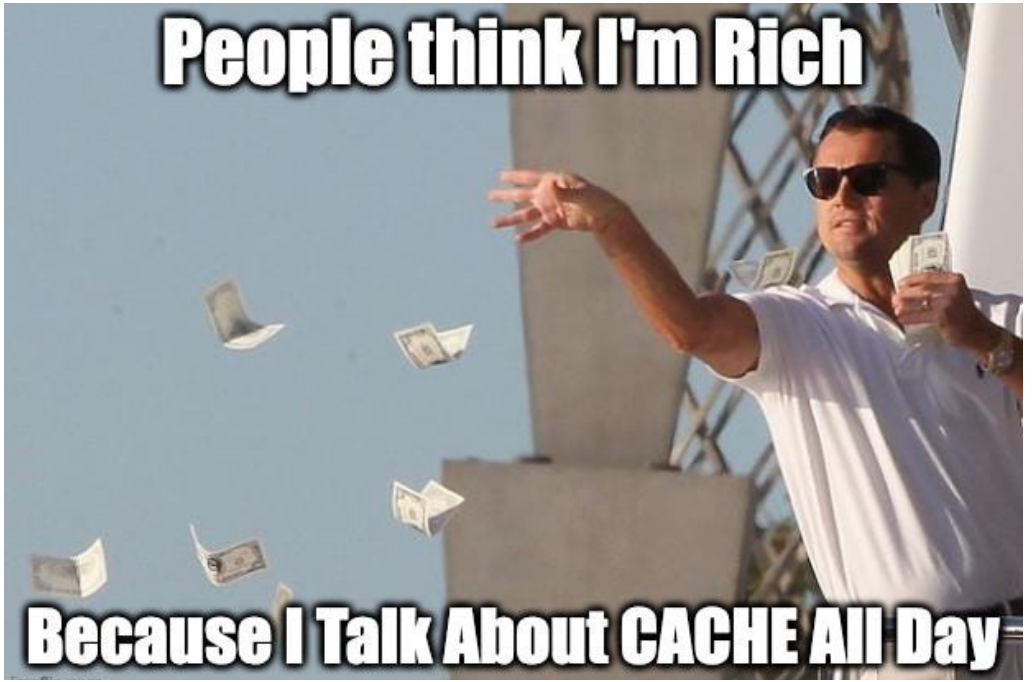
SYNACKTIV

- **Started during a pentest**
  - Audit of the laptops (client wants to know what could be achieved if computer stolen)
  - Bitlocker + TPM but no PIN during boot process
    - Sniffing bitlocker key
    - Decrypt disk to erase the local admin password
  - WHFB installed + Entra ID environment => no mscache

Introduction

SYNACKTIV

- **Can we have an authenticated access on the domain?**
  - Previous users have been authenticated on the domain with the computer => cache file somewhere



WHFB and Microsoft Entra ID

WHFB and Microsoft Entra ID

SYNACKTIV

- **Windows Hello For Business != Windows Hello**
  - WH => Authentication with a Microsoft account or an Identity provider or relying party services supporting Fast ID Online v2.0 authentication.
  - Users can create a PIN or biometric gesture on their personal devices for convenient sign-in.
  - These options make it easier and safer to sign in to computers as it can be backed up for recovery with a Microsoft account.

WHFB and Microsoft Entra ID

SYNACKTIV

- **Authentication types**
  - Windows Hello Face
  - Windows Hello Fingerprint
  - Windows Hello PIN
  - Physical security key
  - …

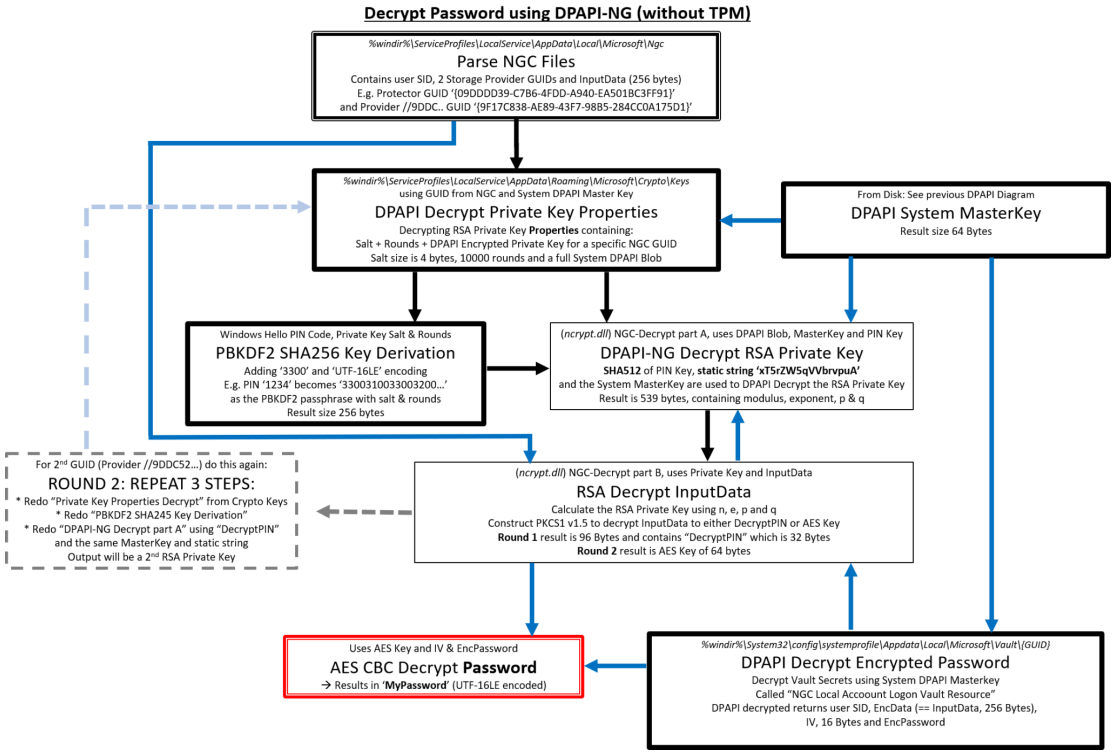
WHFB and Microsoft Entra ID

SYNACKTIV

- **Windows Hello**
  - If you want to retrieve a password when someone is using WH:
    - The SAM hive is no longer used.
    - The final goal is to decrypt a file containing the user password. The file can be a vault or directly a registry key: *HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Authentication\LogonUI\NgcPin\Credentials\S-1-5-21-xxx\encryptedPassword*.

WHFB and Microsoft Entra ID

SYNACKTIV



- https://www.insecurity.be/blog/2020/12/24/dpapi-in-depth-with-tooling-standalone-dpapi/

WHFB and Microsoft Entra ID

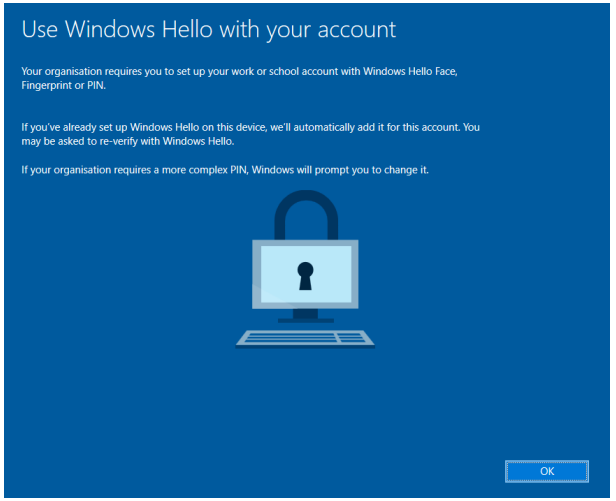
SYNACKTIV

- **Windows Hello For Business != Windows Hello**
  - WHFB => Authentication with a Microsoft Entra ID account, an Active Directory account or an IdP / RP
  - Uses key-based or certificate-based authentication.

WHFB and Microsoft Entra ID

SYNACKTIV

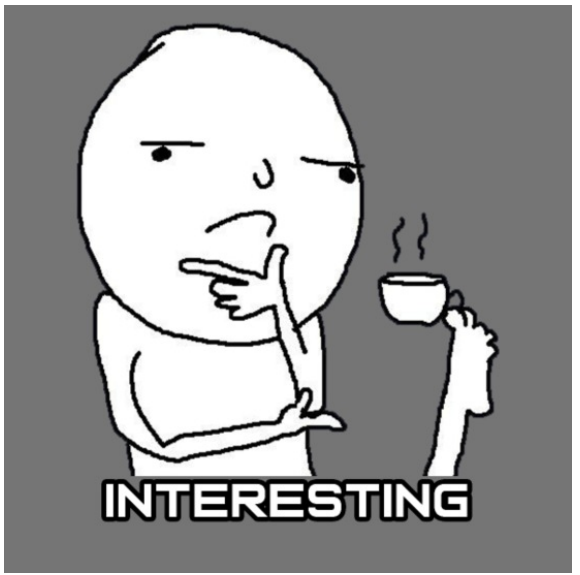
- **Registration process WHFB / Entra ID**
  - After joining a Microsoft Entra ID tenant, reboot and registration process.
  - Creation of a PIN
  - Public/Private key generated, and the PIN is an entropy used to protect the private key



WHFB and Microsoft Entra ID

SYNACKTIV

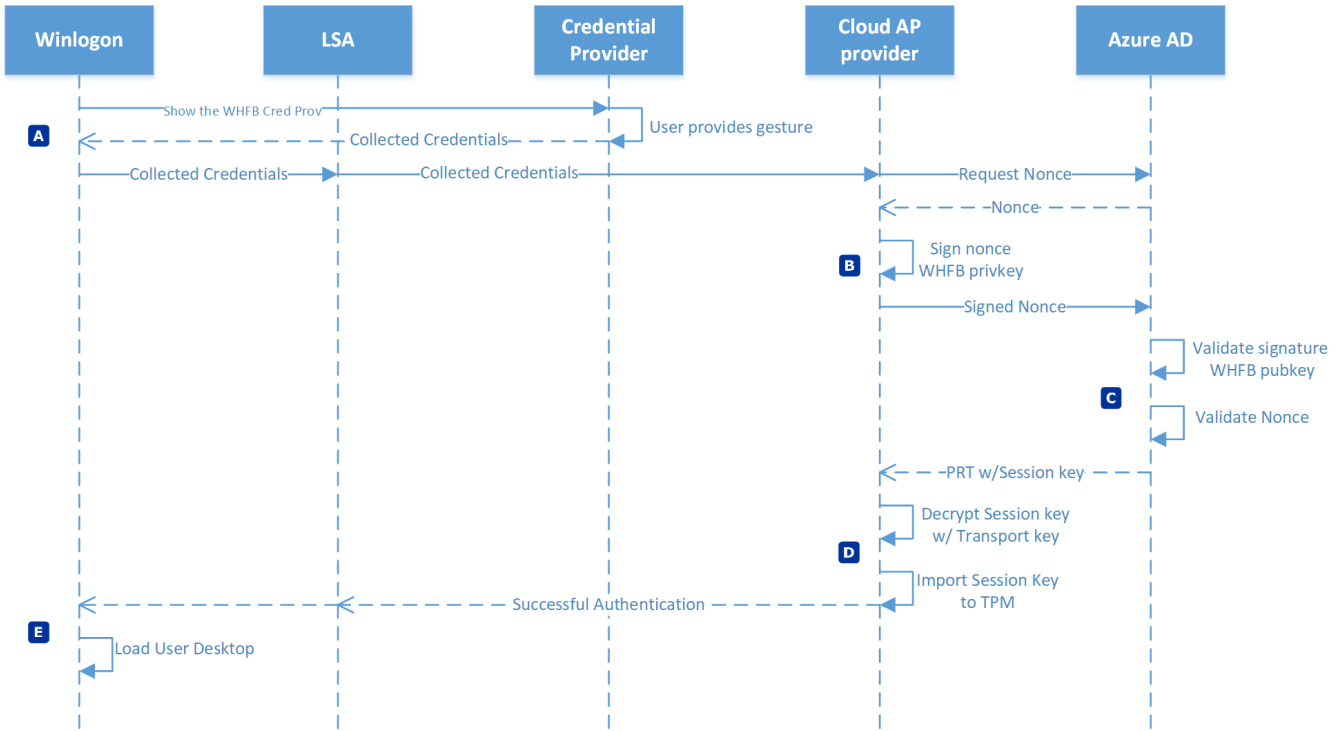
- **Registration process WHFB / Entra ID**
  - TPM: PIN used to access the private key stored in the TPM
    => tamper protection of the TPM provided.
  - Without TPM: same process, but everything is on the filesystem
    => no tamper protection



WHFB and Microsoft Entra ID

SYNACKTIV

- Authentication process WHFB / Entra ID with a TPM



https://learn.microsoft.com/en-us/windows/security/identity-protection/hello-for-business/how-it-works-authentication

WHFB and Microsoft Entra ID

SYNACKTIV

- **Authentication process WHFB / Entra ID**
  - PRT
    - Key artifact of Microsoft Entra ID authentication.
      - => Can be seen as a TGT in Active Directory. Used to sign in a user on their Entra ID device and connected resources.
    - When a PRT is issued, Entra ID issues an encrypted session key to the device. It is encrypted with the public key of the device.
    - Session key => generation of a derived key => could be used to modify and re-sign PRT cookie. This allows us to use the PRT for as long as it is valid (14 days) on other systems than it was issued on.
    - https://dirkjanm.io/digging-further-into-the-primary-refresh-token/

WHFB and Microsoft Entra ID

SYNACKTIV

- **Authentication process WHFB / Entra ID**
  - Cloud Authentication Provider (CloudAP)
    - Windows Authentication Package enabling users to sign in to Windows using their Entra ID or Microsoft Account.

WHFB and Microsoft Entra ID

SYNACKTIV

- **Authentication process WHFB / Entra ID**
  - cloudAP.dll
    - Lives in *lsass.exe* process' memory
    - Used to authenticate a user logon attempt
    - Implements the *SECPKG_FUNCTION_TABLE* structure
      - Mandatory for a security package
      - *LSA_AP_LOGON_USER_EX2* is set to *LsaApLogonUserEx2*
        - Performs the authentication
    - Not documented by Microsoft
      - (Some structures are documented in LSA Whisperer wiki)

WHFB and Microsoft Entra ID

SYNACKTIV

- **What if Entra ID can not be reached ?**
  - The user is still able to perform local authentication
  - Based on a cache file
  - *LsaApLogonUserEx2* is executed with *LogonType = CachedInteractive*

```
typedef enum _SECURITY_LOGON_TYPE {
/* ... */
#if (_WIN32_WINNT >= 0x0501)
        RemoteInteractive, // Remote, yet interactive. Terminal server
        CachedInteractive, // Try cached credentials without hitting the net.
// The types below only exist in Windows Server 2003 and greater
#endif
/* ... */
}
```

Cache data format

Cache data format

SYNACKTIV

- **CacheData file**
  - *%SYSTEM32%\config\systemprofile\AppData\local\microsoft\windows\CloudAPCache\AzureAD\<unique_hash>\Cache\CacheData*
- **Admin privileges needed to read it**

Cached data format

SYNACKTIV

- **One Cache folder per EntraID user**
  - Hash found by browsing HKLM hive and checking keys which contain user information:
    - HKLM:\SOFTWARE\Microsoft\IdentityStore\LogonCache\B16898C6-A148-4967-9171-64D755DA8520\Name2Sid
- **Accessed by cloudAP.dll, within lsass.exe**

Cached data format

SYNACKTIV

- **Why is it an interesting file ?**
  - May be used to retrieve the password and the PIN
    - Implies bruteforce attack
      - More on this later…
  - May be used to obtain the PRT + User DPAPI CredKey
    - For both PIN and password
    - Only if credentials have been bruteforced successfully
      - Limited for PIN if there is a TPM

Cached data format

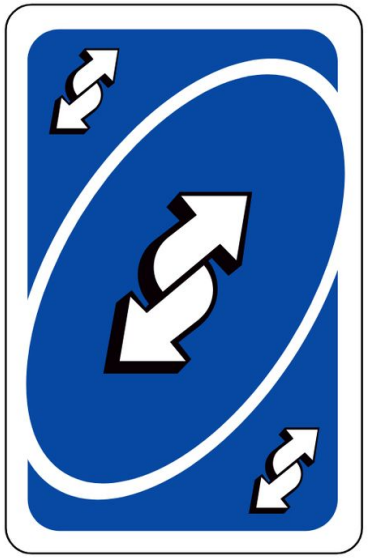
SYNACKTIV

- **Previous work**
  - CacheData PRT decryption when using password
    - PRT_Utils.ps1 from AADInternals repository
  - Windows Hello Ngc PIN Decryption using DPAPI
    - ngccryptokeysdec.py from dpapilab-ng repository
- **Our contribution**
  - CacheData PRT decryption for password and PIN authentication
  - Increase comprehension on the CacheData file format
  - Python script for bruteforcing PIN (without a TPM) or password (with or without a TPM)

Cached data format

SYNACKTIV

- **Methodology**
  - Need to reverse-engineer *cloudAP.dll* to understand authentication process and interaction with the cache file



Cached data format

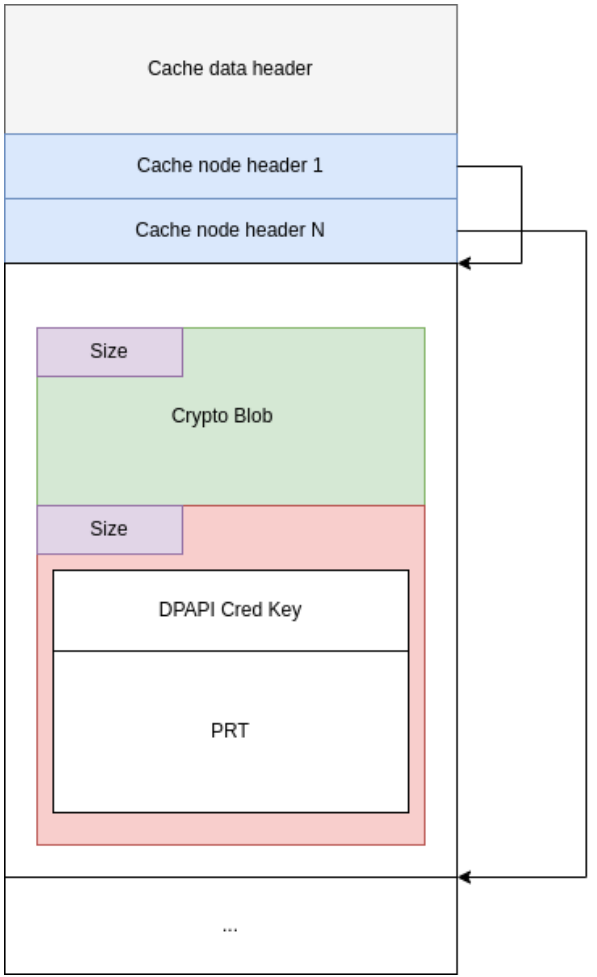
SYNACKTIV

- **Methodology**
  - Static analysis
    - IDA + HexRays Decompiler
    - Public PDB is available :)
      - Common for Microsoft built-in DLLs
      - Functions and global variables are named
  - Dynamic analysis
    - Creation of a Time Travel Debugging (TTD) trace of lsass.exe using Windbg
      - Same trace can be shared among different users !
      - Each DLLs can be easily extracted for static analysis

Cached data format

SYNACKTIV

- **Simple file format**
  - Header with version number, GUID, sha256, number of nodes…
  - Nodes headers (1 or more)
    - Type of node, size of CryptoBlob, size of EncryptedBlob
  - Nodes (1 or more)
- **One node per authentication means**
  - e.g: PIN + password → 2 nodes
  - We only analyzed PIN and password auth



Cached data format

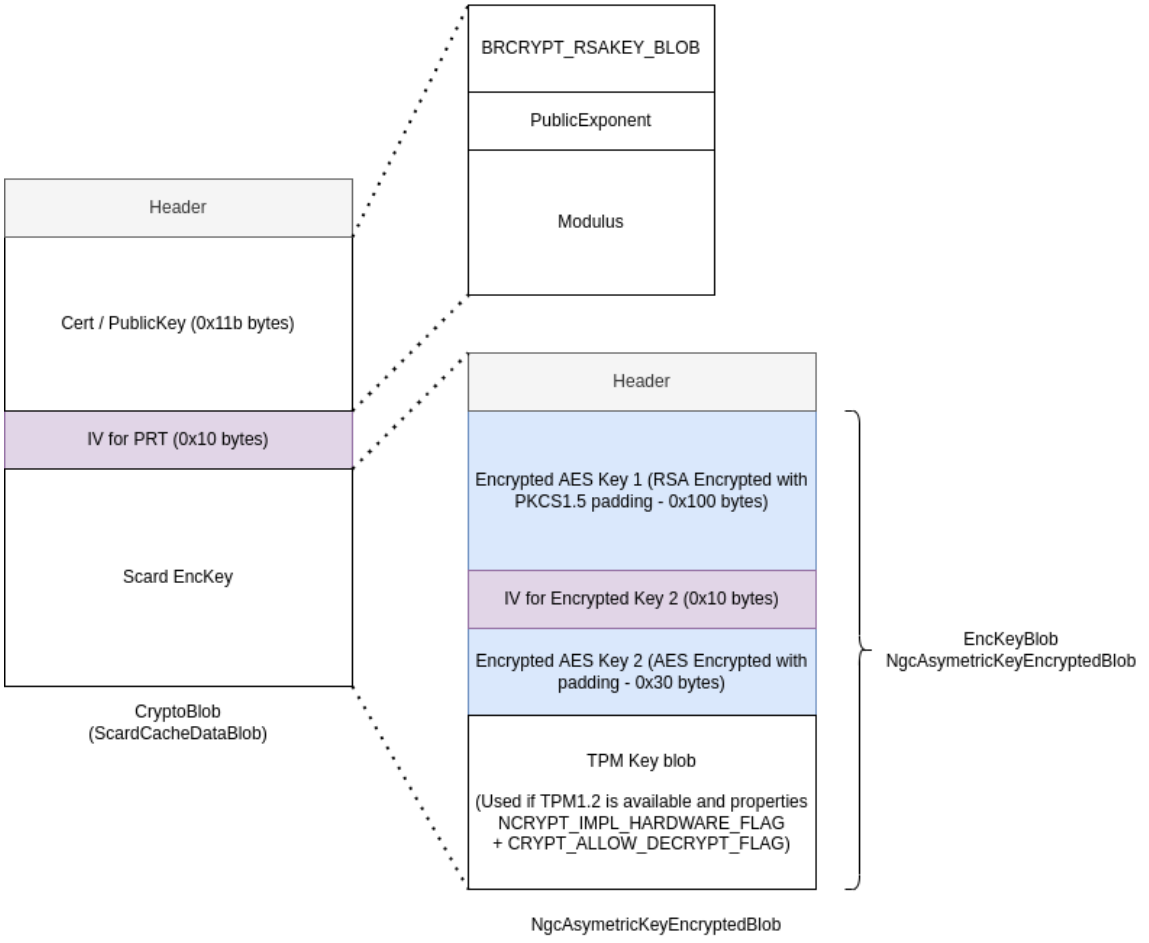
SYNACKTIV

- Parsed by *cloudAP!DeserializeCloudAPCache*
  - *Populates a structure of type CloudAPCache*
    - *With a pointer to an array of structures of tagCacheNodeIdentifier*
- *tagCacheNodeIdentifier*
  - *Node type*
    - *0x1 : Password based authentication*
    - *0x5 : Pin based authentication*
  - *Pointer to + size of CryptoBlob*
    - *Format changes according to node type*
  - *Pointer to + size of EncryptedBlob*
    - *Encrypted PRT + DPAPI CredKey*

Cached data format

SYNACKTIV

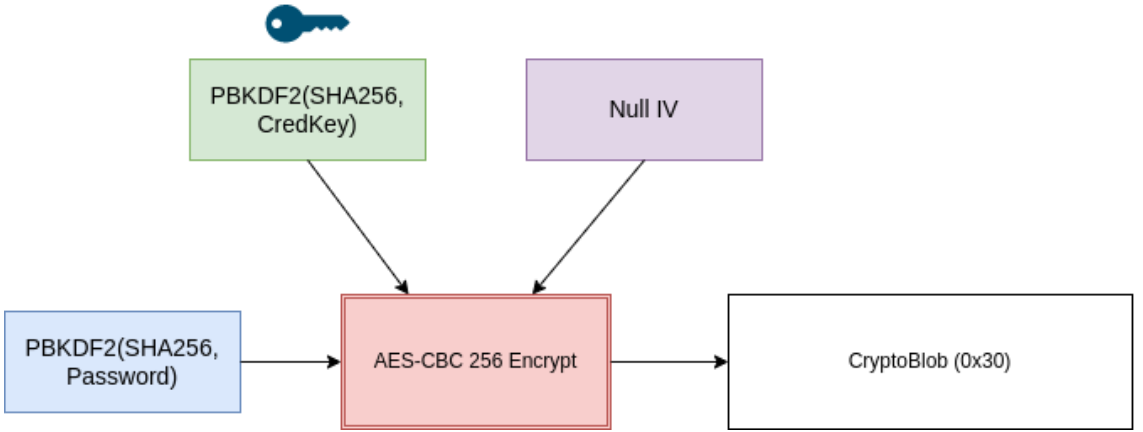
- **CryptoBlob when using PIN based authentication**



Cached data format

SYNACKTIV

- CryptoBlob when using password based authentication



Cached data format

SYNACKTIV

- **DPAPI CredKey**
  - *Stored in a struct CREDENTIAL_KEY (0x60 bytes)*
    - *GUID*
    - *Key stored in a 0x40 bytes buffer*
  - Derived to decrypt latest Master Key
    - *HMAC(SHA1(CredKey), USERSID_UTF16_LE, SHA1)*
- **Stored encrypted in the CacheData**
- **Protected in memory using *LSA_PROTECT_MEMORY* callback**
  - Symetric key generated in *lsasrv!LsaInitializeProtectedMemory*

Cached data for offline authentication

Password

Cached data for offline authentication

SYNACKTIV

- **Password based authentication**
  - PRT is encrypted with an AES key derived from the password
  - Analysing *cloudAP!DeriveKeyFromSecret* function:
    - Takes the password as an argument to create a key:
      - PBKDF2HMAC(SHA256, pwd, lengthkey=32 bytes) without salt iterating 10 000 times as for DPAPI encryption.
  - Resulting key is used in AES-CBC decryption (with a null IV) on the CacheData encryptedBlob

PIN authentication

Cached data for offline authentication

SYNACKTIV

- **Introduction - With a PIN**
  - NGC (or DPAPI-NG) used:
    - Next-Gen-Cryptography → long-term replacement for the Microsoft CryptoAPI
    - Provides a set of APIs that can be used to easily encrypt and decrypt content to authorization principals across multiple computers
    - Works with providers, protectors and items:
      - Provider: component responsible for managing cryptographic operations and interacting with the NGC framework. Two different types of providers: Key Storage provider or Cryptographic Service Provider
      - Protector: method or technique used to encrypt and protect sensitive data.

Cached data for offline authentication

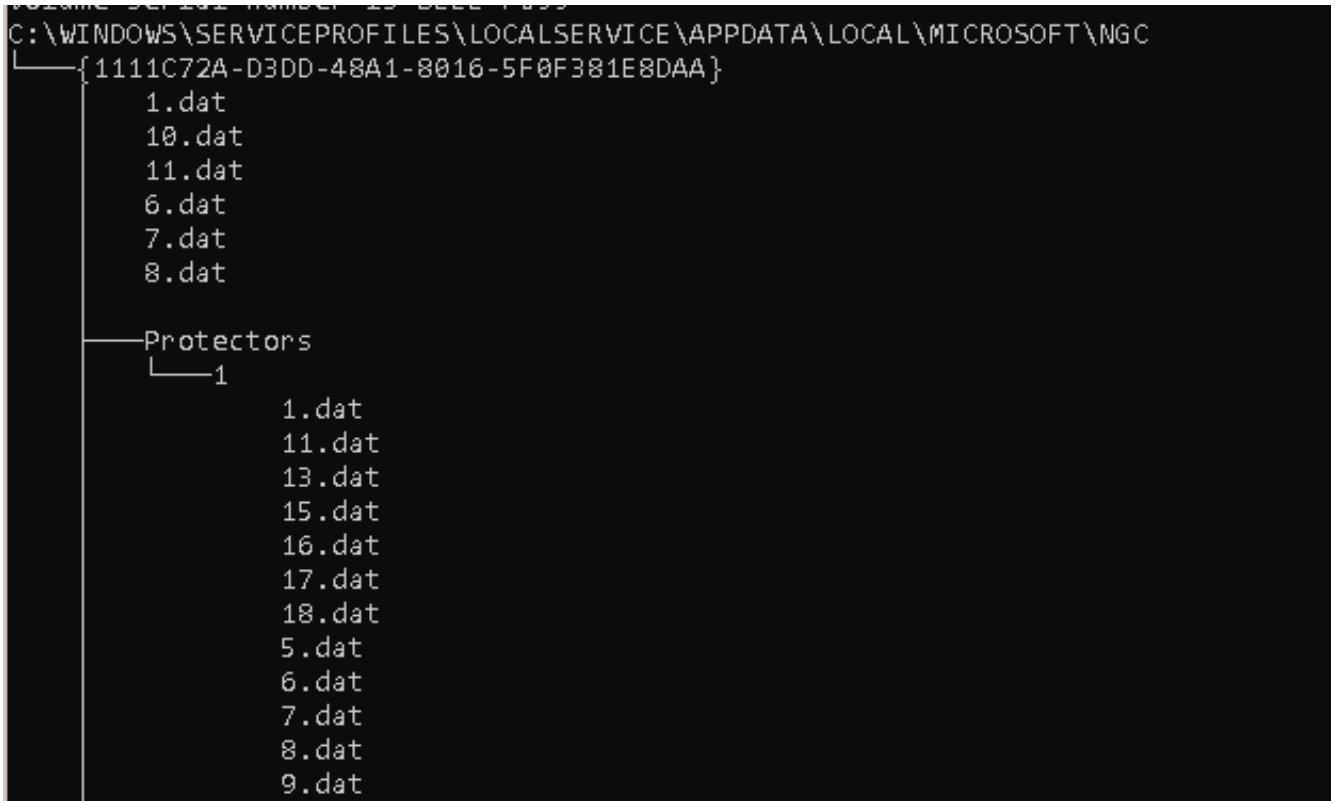
SYNACKTIV

- **Introduction - With a PIN**
  - NGC
    - Location: %windir%\ServiceProfiles\LocalService\AppData\Local\Microsoft\Ngc
    - System privileges needed to access it
    - Protectors, providers and items metadata can be retrieved by parsing non-encrypted data inside it.

Cached data for offline authentication

SYNACKTIV

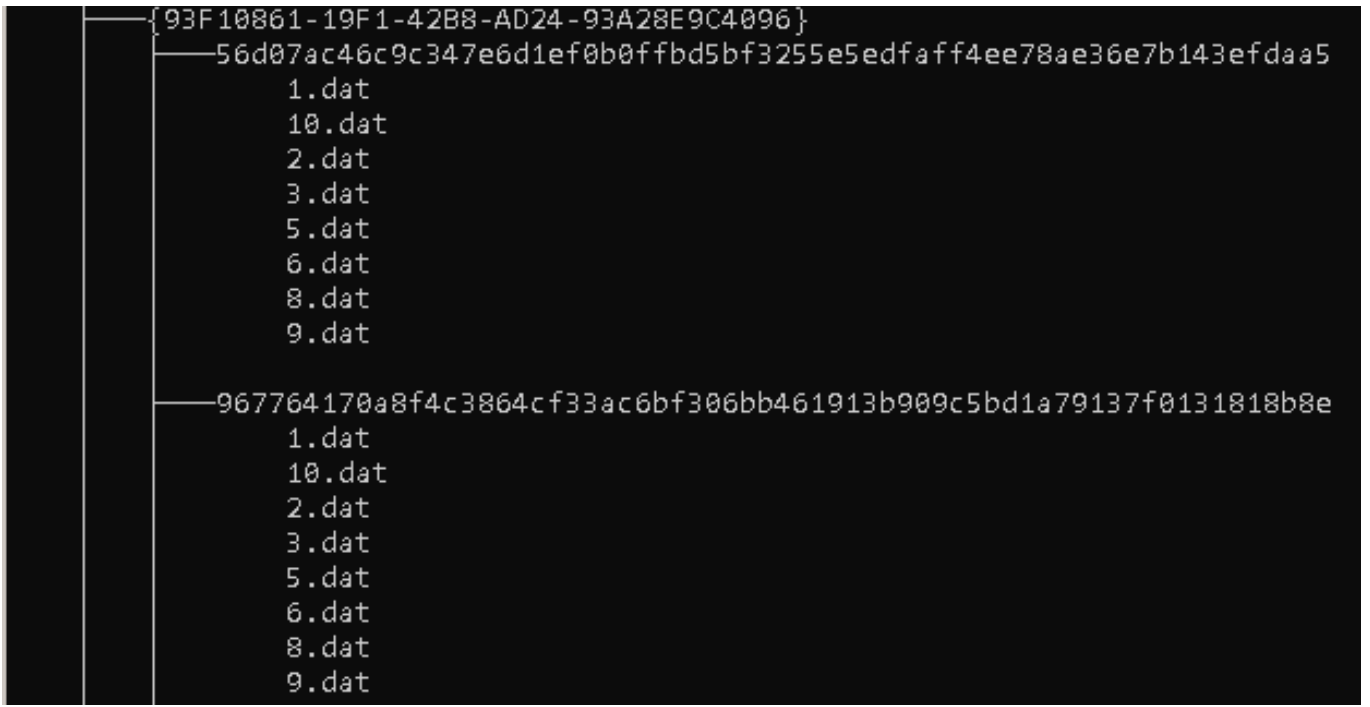
- **Introduction - With a PIN**
  - NGC



Cached data for offline authentication

SYNACKTIV

- **Introduction - With a PIN**
  - NGC



Cached data for offline authentication

SYNACKTIV

- **Introduction - With a PIN**
  - Inside the root NGC folder:
    - NGC GUID folder which contains
      - User SID (1.dat)
      - Main provider (7.dat)

```
[!] Parsing the Ngc folder
[+] NGC GUID      : {BD1E1811-FFFB-4F76-850E-03DDF974D27E}
[+] User SID      : S-1-12-1-1473278482-1076885373-2432020880-3020655032
[+] Main Provider : Microsoft Software Key Storage Provider
```

https://github.com/tijldeneut/dpapilab-ng

Cached data for offline authentication

SYNACKTIV

- **Introduction - With a PIN**
  - Key Storage Providers:
    - With TPM → Microsoft Plaftform Crypto Provider → Protector stored in the TPM chip
    - Without TPM → Microsoft Software Key Storage Provider → Protector stored locally
    - Other providers exist with NGC: SmartCard Key Storage provider…

```
$ python3 _ngc_step_by_step_on_and_offline.py <ngc_folder>
[...]
== Protectors ==
[-] Protector "1" is being stored in the TPM chip.
= 1 =
[+] Provider   : Microsoft Platform Crypto Provider
```

```
$ python3 _ngc_step_by_step_on_and_offline.py <ngc_folder>
[...]
== Protectors ==
= 1 =
[+] Provider   : Microsoft Software Key Storage Provider
```

https://github.com/tijldeneut/dpapilab-ng

Cached data for offline authentication

SYNACKTIV

- **Introduction - With a PIN**
  - Protectors:
    - NGC\<NGC_GUID>\Protectors\1\:
      - 1.dat → Name of the protector
      - 2.dat → Key GUID of the protector (missing if stored on the TPM)
      - 15.dat → Encrypted data

```
== Protectors ==
= 1 =
[+] Provider  : Microsoft Software Key Storage Provider
[+] Key Name  : {F4945423-90A3-4764-929A-314097175160}
[+] Timestamp : 2024-02-07 21:23:08
[+] Data Size : 256 byte(s)
```

https://github.com/tijldeneut/dpapilab-ng

Cached data for offline authentication

SYNACKTIV

- **Introduction - With a PIN**
  - Items:
    - NGC\<NGC_GUID>\<GUID>\:
      - Each item is stored inside a folder
      - 1.dat → Name of the item
      - 2.dat → Provider name

```
* 967764170a8f4c3864cf33ac6bf306bb461913b909c5bd1a79137f0131818b8e
[+] Name     : //CA00CFA8-EB0F-42BA-A707-A3A43CDA5BD9
[+] Provider : Microsoft Software Key Storage Provider
[+] Key Name : {9B6DC1EA-F6CC-46AB-8226-67808A8494F1}
```

https://github.com/tijldeneut/dpapilab-ng

Cached data for offline authentication

SYNACKTIV

- Introduction - With a PIN

PIN without TPM

Cached data for offline authentication

SYNACKTIV

- **Authentication with a PIN without a TPM**
  - A first RSA private key needs to be constructed from encrypted data (called a BCRYPT RSA Private Key Blob)
  - Used to decrypt and obtain a DecryptPIN, also used to obtain a second RSA private key constructed from another BCRYPT RSA Private Key Blob.

Cached data for offline authentication

SYNACKTIV

- **Authentication with a PIN without a TPM**
  - Blobs are stored inside %windir%\ServiceProfiles\LocalService\AppData\Roaming\Microsoft\Crypto\Keys, identified by cleartext metadata for each key.

```
> hexdump -C 445ae139ebd246ac6410b2292735fe52_c2e570f7-a2b1-4483-b686-ab4ab03d
00000000  01 00 00 00 00 00 00 00  4c 00 00 00 01 00 03 00  |........L.......|
00000010  5b 01 00 00 72 02 00 00  0c 03 00 00 00 00 00 00  |[...r...........|
00000020  00 00 00 00 00 00 00 00  00 00 00 00 7b 00 39 00  |............{.9.|
00000030  38 00 39 00 43 00 46 00  31 00 41 00 38 00 2d 00  |8.9.C.F.1.A.8.-.|
00000040  42 00 38 00 43 00 39 00  2d 00 34 00 30 00 46 00  |B.8.C.9.-.4.0.F.|
00000050  41 00 2d 00 42 00 46 00  42 00 33 00 2d 00 35 00  |A.-.B.F.B.3.-.5.|
00000060  31 00 43 00 43 00 37 00  46 00 45 00 38 00 36 00  |1.C.C.7.F.E.8.6.|
00000070  41 00 41 00 31 00 7d 00  2c 00 00 00 00 00 00 00  |A.A.1.},........|
00000080  00 00 00 00 10 00 00 00  08 00 00 00 4d 00 6f 00  |............M.o.|
00000090  64 00 69 00 66 00 69 00  65 00 64 00 ba de c9 b9  |d.i.f.i.e.d.....|
000000a0  af 4e da 01 2f 01 00 00  0a 00 00 00 00 00 00 00  |.N../...........|
000000b0  00 00 00 00 1b 01 00 00  52 53 41 31 00 08 00 00  |........RSA1....|
000000c0  03 00 00 00 00 01 00 00  00 00 00 00 00 00 00 00  |................|
```

Cached data for offline authentication

SYNACKTIV

- **Authentication with a PIN without a TPM**
  - The first BCRYPT RSA Private Key Blob linked with NGC is decrypted using DPAPI mechanisms after multiple steps involving:
    - System masterkeys
    - GUID of the Protector
    - SYSTEM and SECURITY hives
    - The PIN <= **Bruteforce here**
    - Static entropy strings
  - A BCRYPT RSA private key is constructed using this blob (contains the Modulus, the exponent, prime1 and prime2).

    https://www.insecurity.be/blog/2020/12/24/dpapi-in-depth-with-tooling-standalone-dpapi/

Cached data for offline authentication

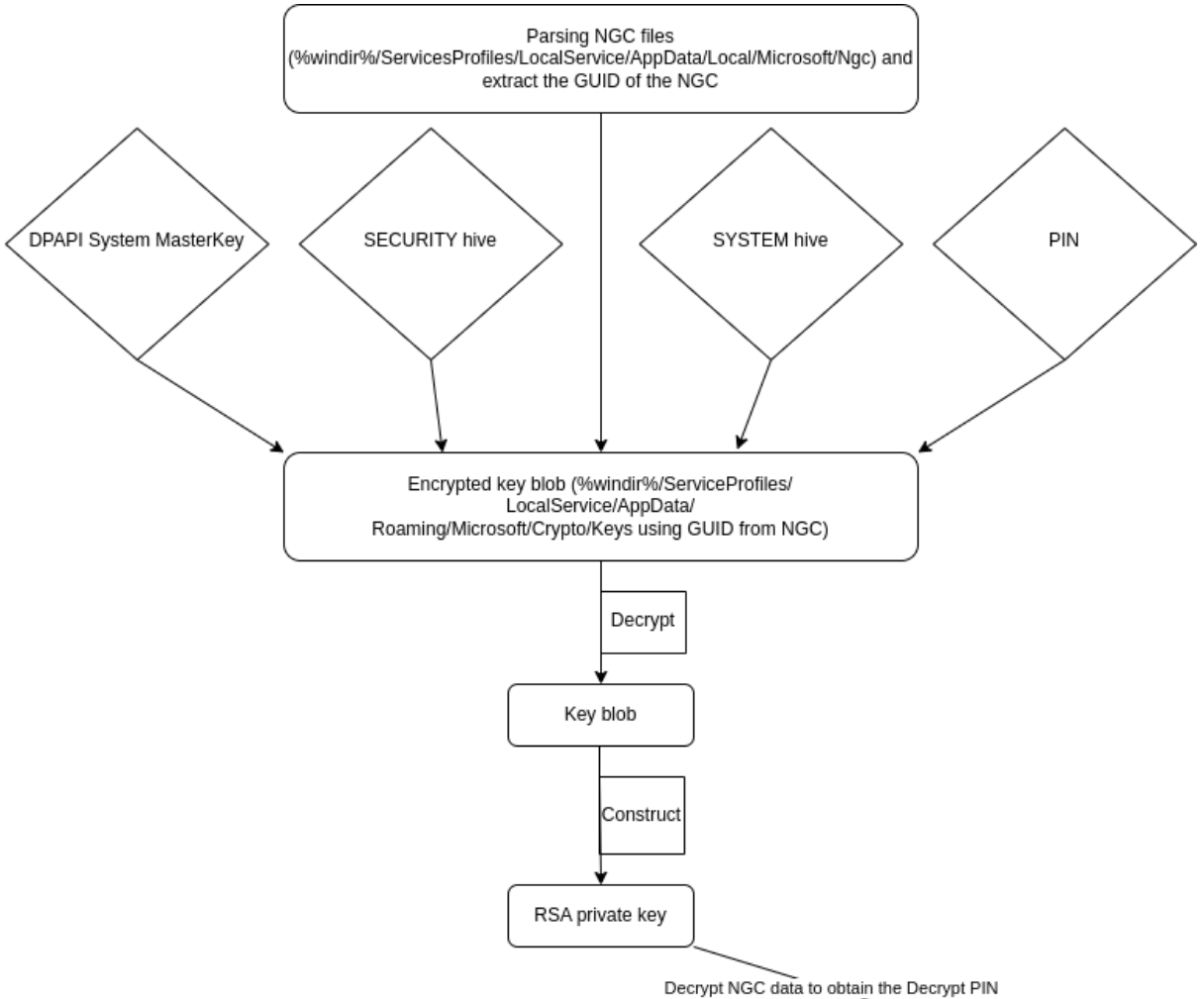
SYNACKTIV

- **Authentication with a PIN without a TPM**
  - This private key is used to decrypt NGC data (15.dat)
    - 3 pins of 32 bits inside
    - The DecryptPIN is the one needed (the second one inside the .dat file)
  - The second BCRYPT RSA Private Key Blob is decrypted using the the Key ID of the item used for WHFB (//CA00CFA8-EB0F-42BA-A707-A3A43CDA5BD9) with the same method replacing the PIN by the DecryptPIN, and the CryptoKey blob by the one linked with the Key ID of the item.
  - Second BCRYPT RSA Private Key constructed (same previous method).

https://www.insecurity.be/blog/2020/12/24/dpapi-in-depth-with-tooling-standalone-dpapi/

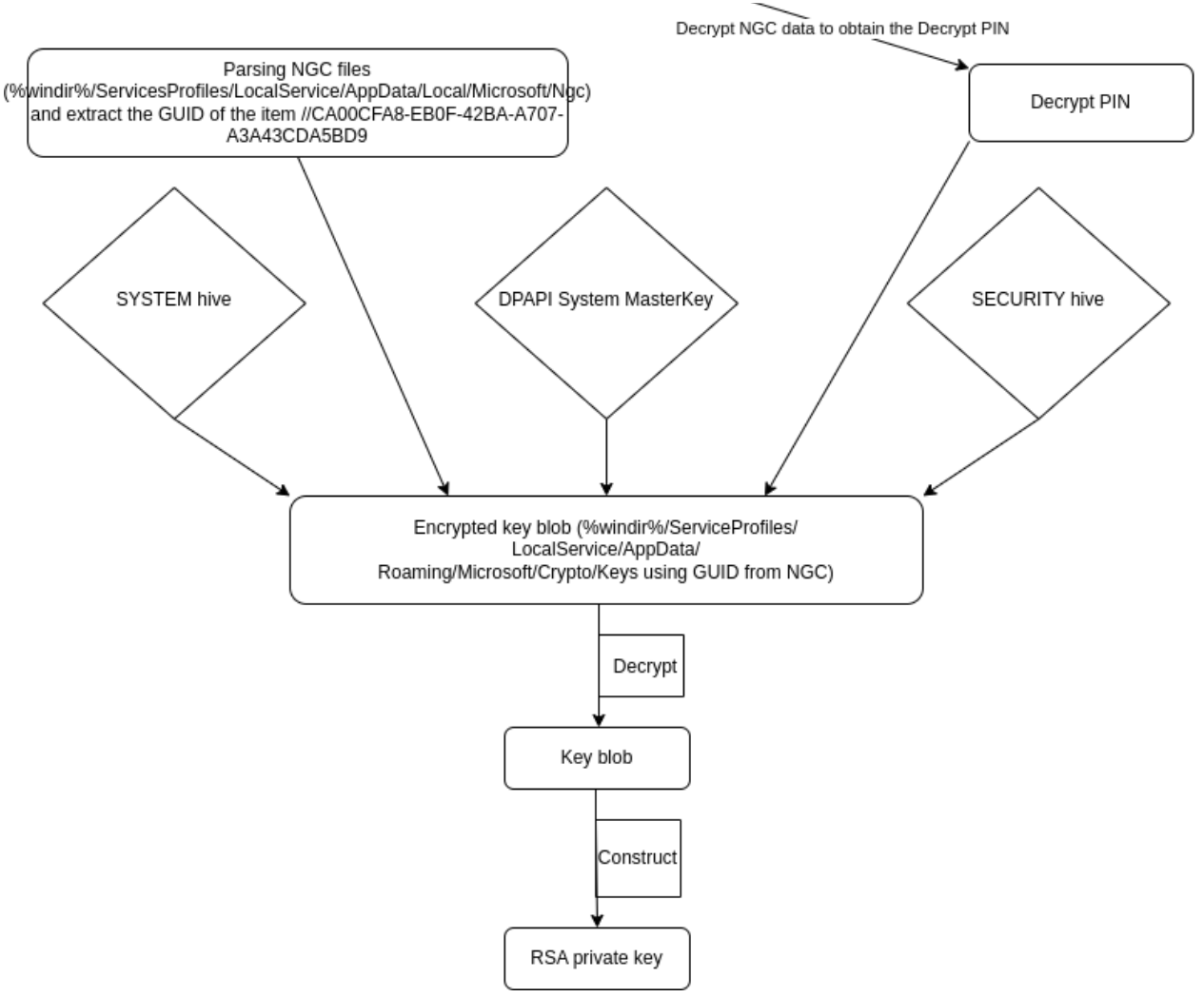
Cached data for offline authentication

SYNACKTIV



Cached data for offline authentication

SYNACKTIV



Cached data for offline authentication

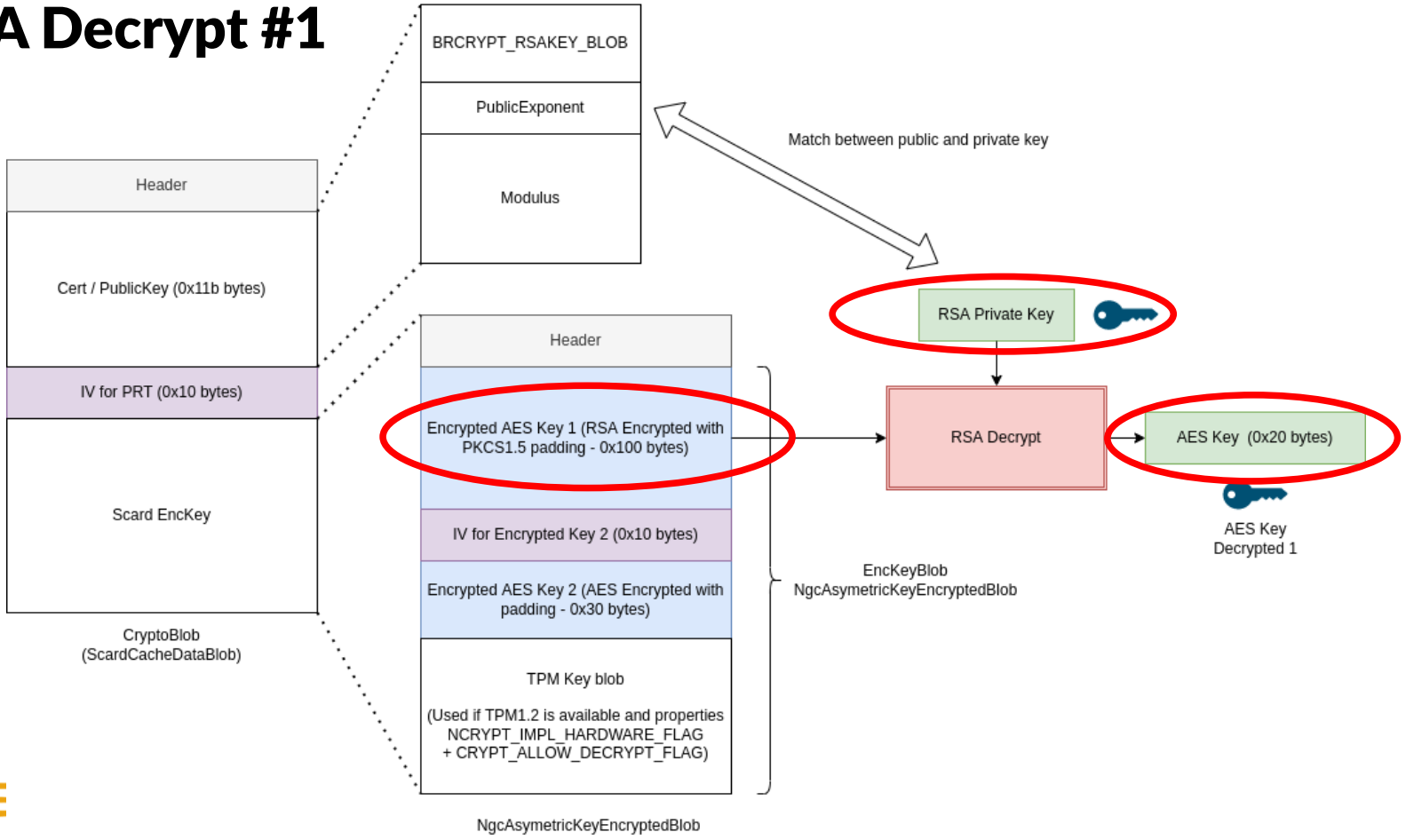
SYNACKTIV

- **PRT decryption with a PIN (no TPM)**
  - The second BCRYPT RSA Private Key is used to encrypt an AES key (Encrypted AES Key 1)
    - Stored encrypted in the CacheData as a blob of 0x100 bytes
      - Big integer of 2048 bits (RSA Encryption + PKCS1.5 padding)

Cached data for offline authentication

SYNACKTIV

- **RSA Decrypt #1**



Cached data for offline authentication

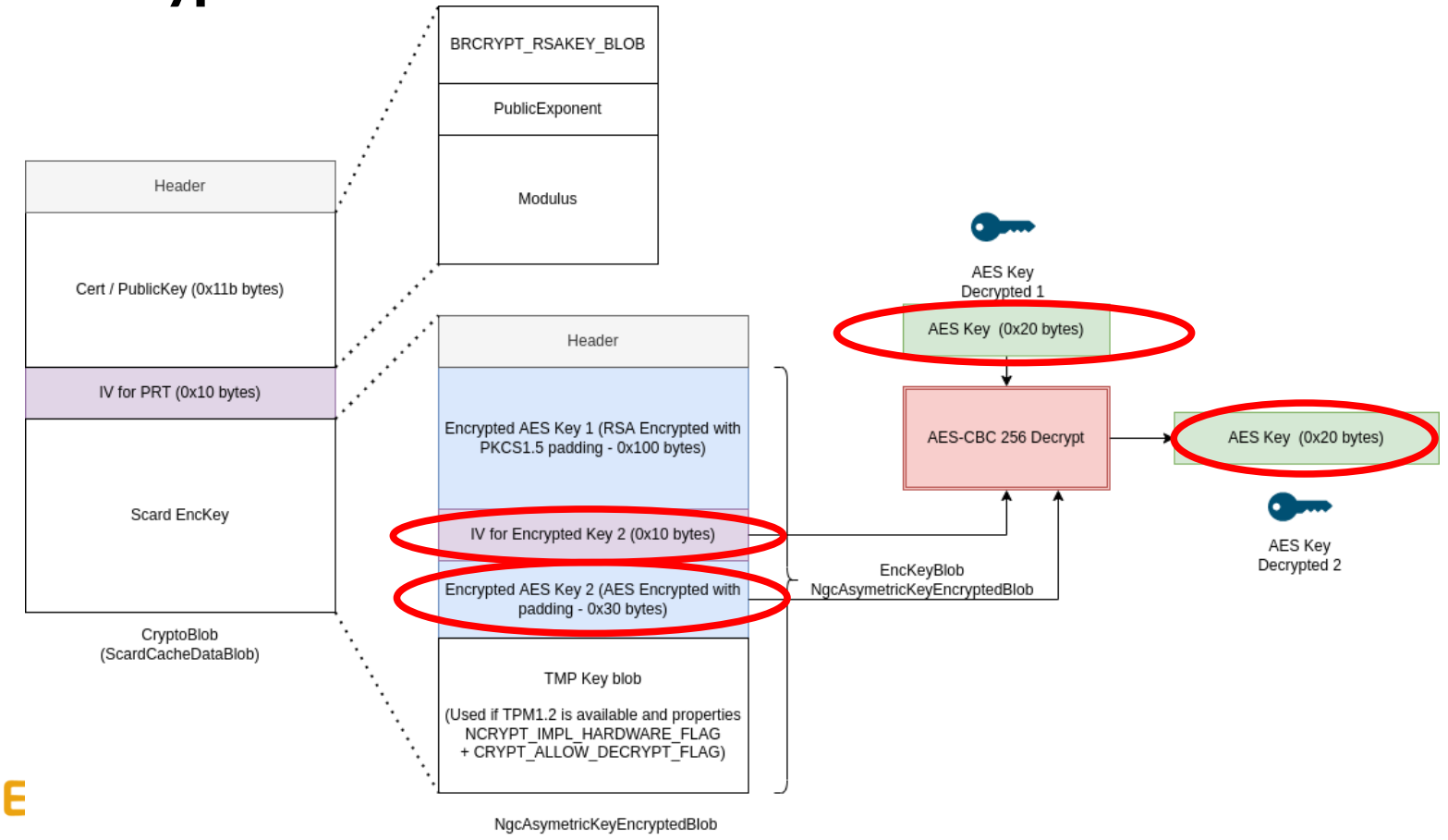
SYNACKTIV

- **PRT decryption with a PIN (no TPM)**
  - This AES key is used to decrypt another AES key (Encrypted AES Key 2)
    - AES-CBC 256 with custom IV in the CacheData
    - Stored in CacheData as a blob of 0x30 bytes (AES-256 bits key + padding)

Cached data for offline authentication

SYNACKTIV

- **AES Decrypt #1**



Cached data for offline authentication

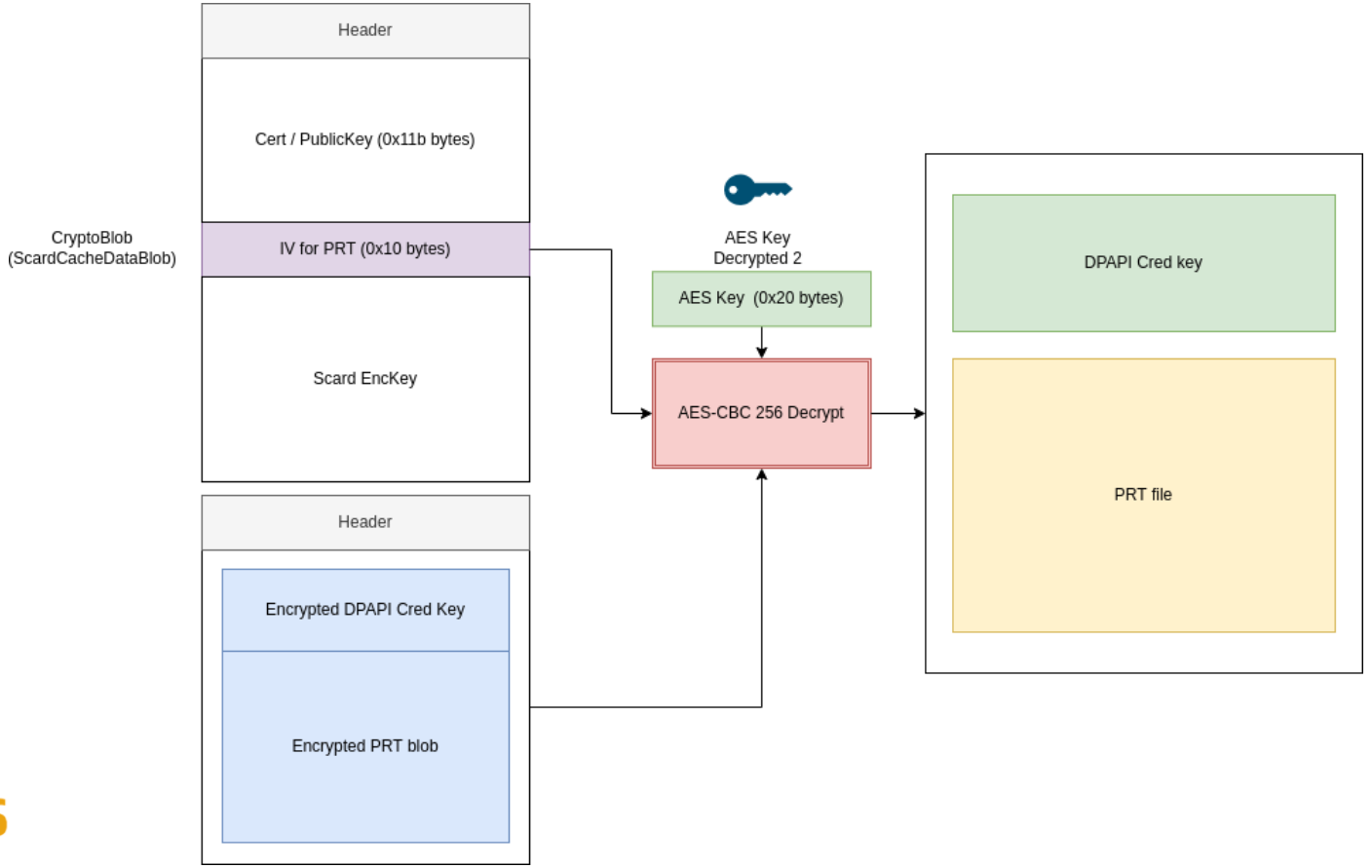
SYNACKTIV

- **Authentication with a PIN without a TPM**
  - The final AES key is used to decrypt the PRT + DPAPI Cred key
    - AES-CBC 256 with (another) custom IV in the CacheData

Cached data for offline authentication

SYNACKTIV

- AES Decrypt #2



PIN with TPM

Cached data for offline authentication

SYNACKTIV

- **Using a PIN, the format of the CacheData *cryptoBlob* is the same with a TPM**
- **Authentication with a PIN with a TPM**
  - Private key is stored on the TPM + PIN is used to access it → bruteforce very hard
  - TPM 1.2: protection implemented by the manufacturer → TPM chips were not equal regarding the mechanism in place.
  - TPM 2.0: TPM configured by Windows to lock after 32 authorization failures and to forget one authorization failure every 10 minutes.

Cached data for offline authentication

SYNACKTIV

- Authentication with a PIN with a TPM

```
PS C:\Windows\system32> Get-TPM

TpmPresent                : True
TpmReady                  : True
TpmEnabled                : True
TpmActivated              : True
TpmOwned                  : True
RestartPending            : False
ManufacturerId            : 1229081856
ManufacturerIdTxt         : IBM
[...]
LockedOut                 : False
LockoutHealTime           : 10 minutes
LockoutCount              : 0
LockoutMax                : 31
SelfTest                  : {}


PS C:\Windows\system32> tpmtool getdeviceinformation

-TPM Present: True
-TPM Version: 2.0
[...]
```

Demo

Demo

- https://github.com/synacktiv/CacheData_decrypt
- PR are welcome :)

A Word on DPAPI

A word on DPAPI

SYNACKTIV

- **This Data Protection API (DPAPI) is a pair of function calls (CryptProtectData / CryptUnprotectData) that provide operating system-level data protection services to user and system processes.**

- **DPAPI Master Key :**
  - DPAPI generates a strong key called the MasterKey.
  - The MasterKey is a “strong secret”
  - It is used to generate the Session Key used for encryption

A word on DPAPI

SYNACKTIV

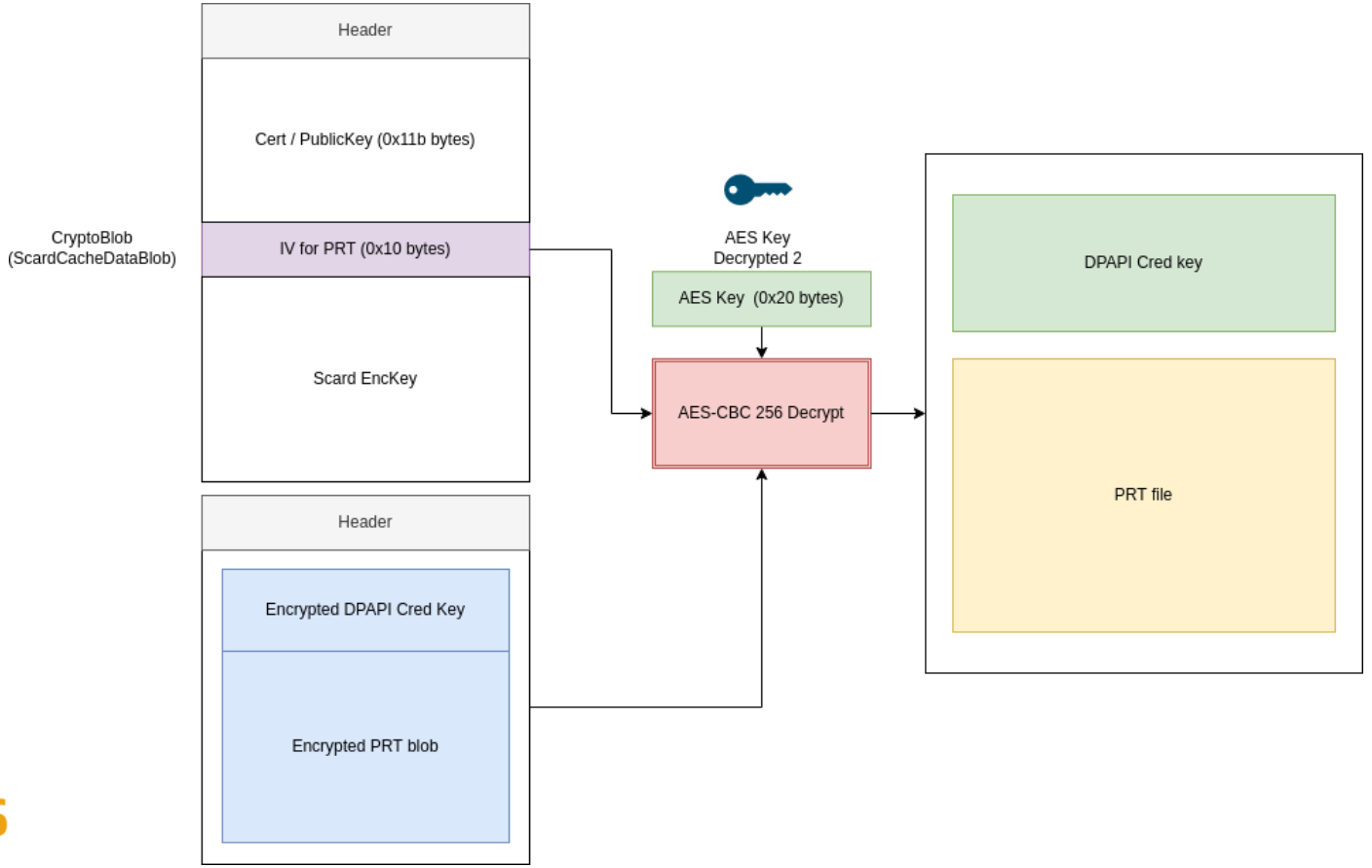
- **DPAPI Master Key :**

Local User	Domain User	SYSTEM
PBKDF2 (SHA1/(Pwd))	PBKDF2 (NTHASH)	Own MasterKey

A word on DPAPI

SYNACKTIV

- What is the DPAPI Cred Key ?



A word on DPAPI

SYNACKTIV

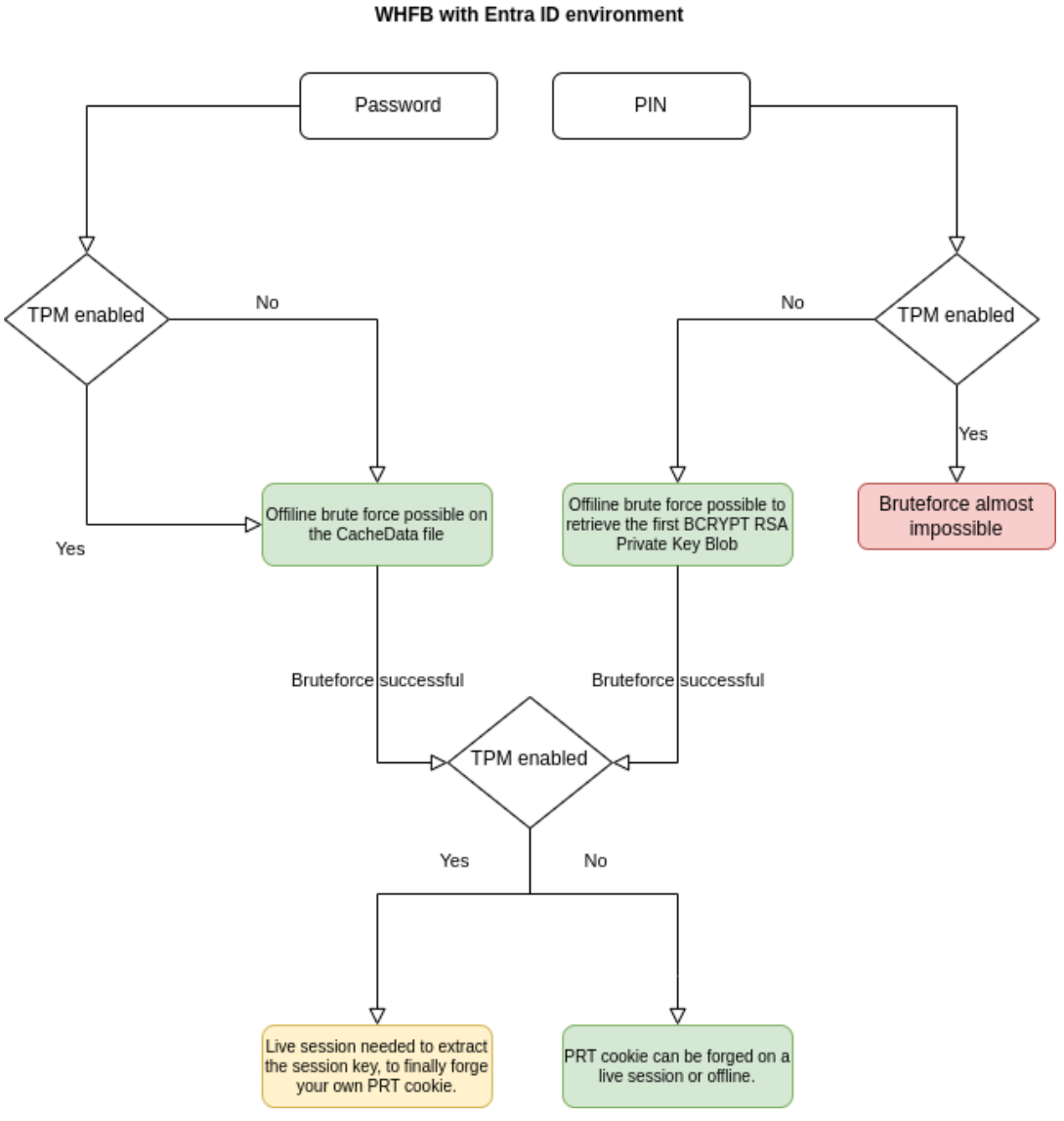
- In an Entra ID environment, the DPAPI masterkeys is not derived from the password of the user

- The CredKey is derived as follows and serves as the base secret to create the DPAPI masterkeys of the user: HMAC(SHA1(CredKey), USERSID_UTF16_LE, SHA1)

- A PR To Diana was made to integrate it thanks to @l4x4

Conclusion

Conclusion


WHFB with Entra ID environment
Password
PIN
TPM enabled
No
No
TPM enabled
Yes
Yes
Offiline brute force possible on the CacheData file
Offiline brute force possible to retrieve the first BCRYPT RSA Private Key Blob
Bruteforce almost impossible
Bruteforce successful
Bruteforce successful
TPM enabled
Yes
No
Live session needed to extract the session key, to finally forge your own PRT cookie.
PRT cookie can be forged on a live session or offline.

Conclusion

SYNACKTIV

- More research needed to understand how it works when using other means of authentication (SmartCard…)
- The CacheData file is a goldmine
- TPM enhances drastically security for credentials

References

SYNACKTIV

https://learn.microsoft.com/en-us/windows/security/identity-protection/hello-for-business/

https://www.insecurity.be/blog/2020/12/24/dpapi-in-depth-with-tooling-standalone-dpapi/

https://github.com/tijldeneut/diana

https://learn.microsoft.com/en-us/windows/security/identity-protection/hello-for-business/hello-how-it-works-authentication

https://dirkjanm.io/

https://github.com/dirkjanm/ROADtools

References

https://github.com/EvanMcBroom/lsa-whisperer/blob/master/wiki/sspi/cloudap.asciidoc#_cloudap_credkey_info

https://dirkjanm.io/digging-further-into-the-primary-refresh-token/

https://www.synacktiv.com/en/publications/windows-secrets-extraction-a-summary

https://learn.microsoft.com/en-us/windows/win32/seccng/cng-portal

https://github.com/tijldeneut/dpapilab-ng

https://helgeklein.com/blog/checking-windows-hello-for-business-whfb-key-storage-tpm-hardware-or-software/

References

https://learn.microsoft.com/en-us/windows/win32/api/bcrypt/ns-bcrypt-bcrypt_rsakey_blob

https://research.nccgroup.com/wp-content/uploads/2020/07/blackhat_europe_2011_exporting_non-exportable_rsa_keys.pdf

https://github.com/synacktiv/CacheData_decrypt

https://github.com/Gerenios/AADInternals

https://github.com/tijldeneut/dpapilab-ng/blob/main/_ngc_step_by_step_on_and_offline.py

https://github.com/tijldeneut/dpapilab-ng/blob/main/ngccryptokeysdec.py

https://i.blackhat.com/USA21/Wednesday-Handouts/us-21-Tsarfati-Bypassing-Windows-Hello-For-Busniess-And-Pleasure.pdf

SYNACKTIV

https://www.linkedin.com/company/synacktiv

https://twitter.com/synacktiv

https://synacktiv.com