

LeHack 2023

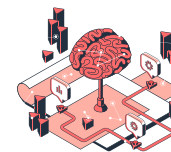
DPAPI – Don't Put Administration Passwords In

login-securite.com

agenda.



C'est quoi
la **DPAPI**?



DPAPI-NG



Point de vue d'un
auditeur



Comment se
protéger ?

Qui sommes-nous.



Pierre-Alexandre
VANDEWOESTYNE
TouF (@T00uF)

CTO
chez Login Sécurité

DonPAPI

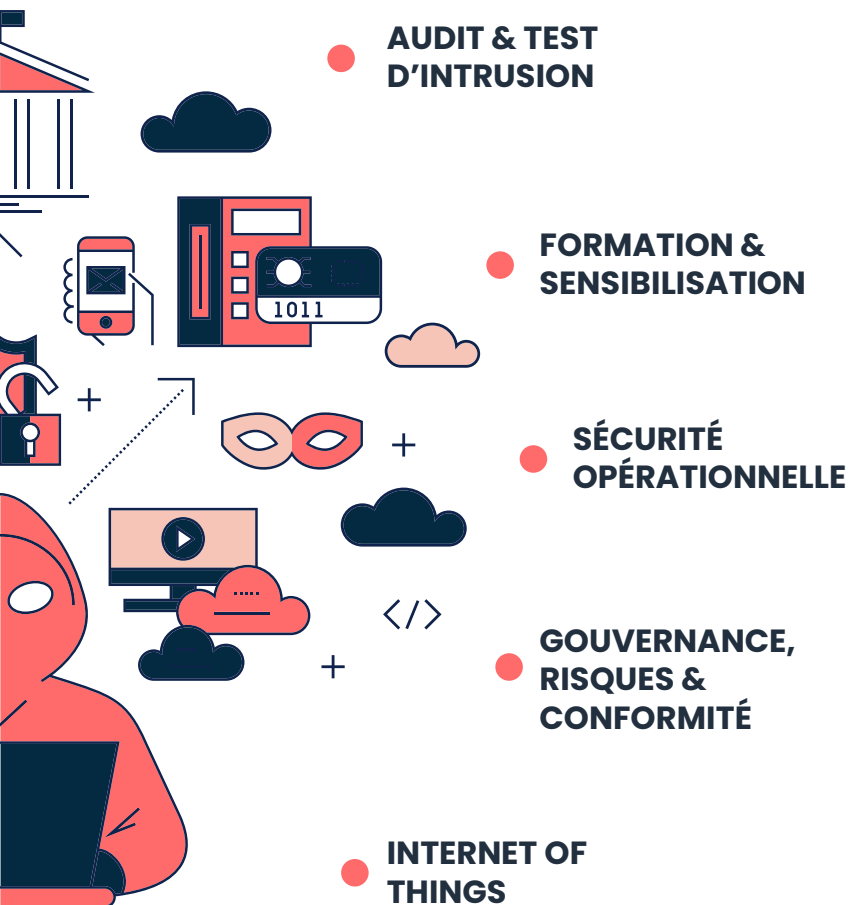


Thomas
SEIGNEURET
zblurx (@_zblurx)

Consultant cybersécurité
chez Login Sécurité

Dploit / Cme --dpapi

Qui sommes-nous.



150 Ti Interne
En 2023



60
collaborateurs

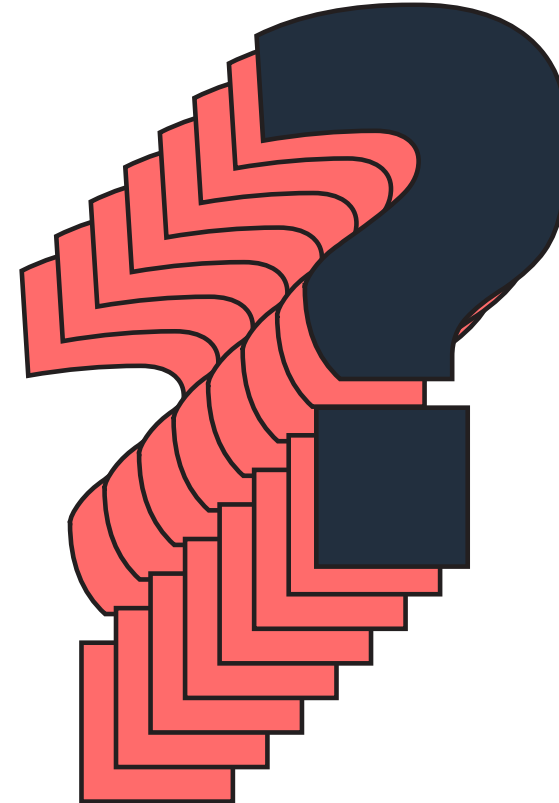


LeHack 2023

DPAPI.

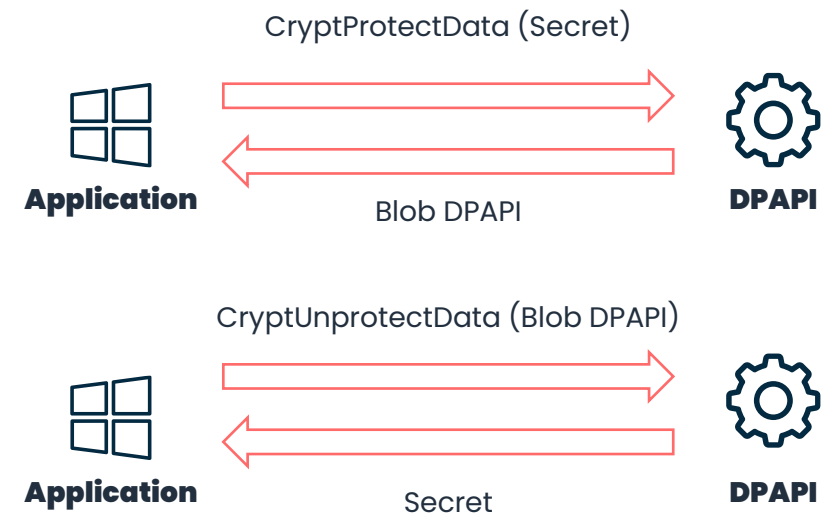
C'est quoi la
DPAPI ?

login-securite.com



Et Microsoft créa ... La Data Protection API.

- DPAPI -> Data Protection API
- Introduit dans **Windows 2000**
- Gestion du chiffrement symétrique des secrets dans un environnement Windows
- L'API nous laisse gérer le stockage du blob chiffré
- Facilite la vie des développeurs :
 - **CryptProtectData** : chiffre la donnée
 - **CryptUnprotectData** : déchiffre la donnée



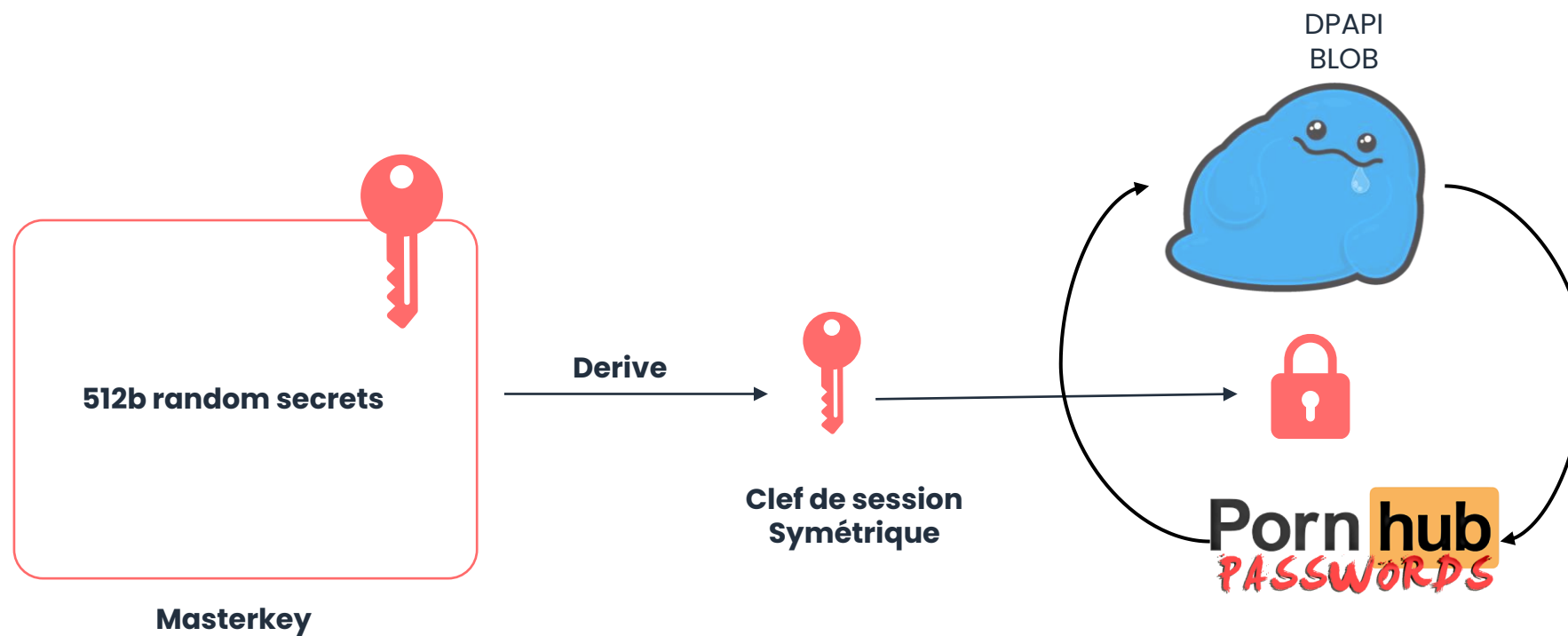
Et Microsoft créa ... La Data Protection API.

Utilisé par Windows (et des applications third party sur Windows) pour chiffrer **toute sorte de secret** :

- Cache de mots de passe des comptes du domaine
- Mots de passe de connexions RDP
- Tâches planifiées
- Mots de passe wifi
- Certificats
- KeePass
- Navigateurs chromium based (Chrome, Edge, Brave)
- Internet Explorer
- Etc.



Dissection de la DPAPI.



Dissection de la DPAPI. – Le BLOB



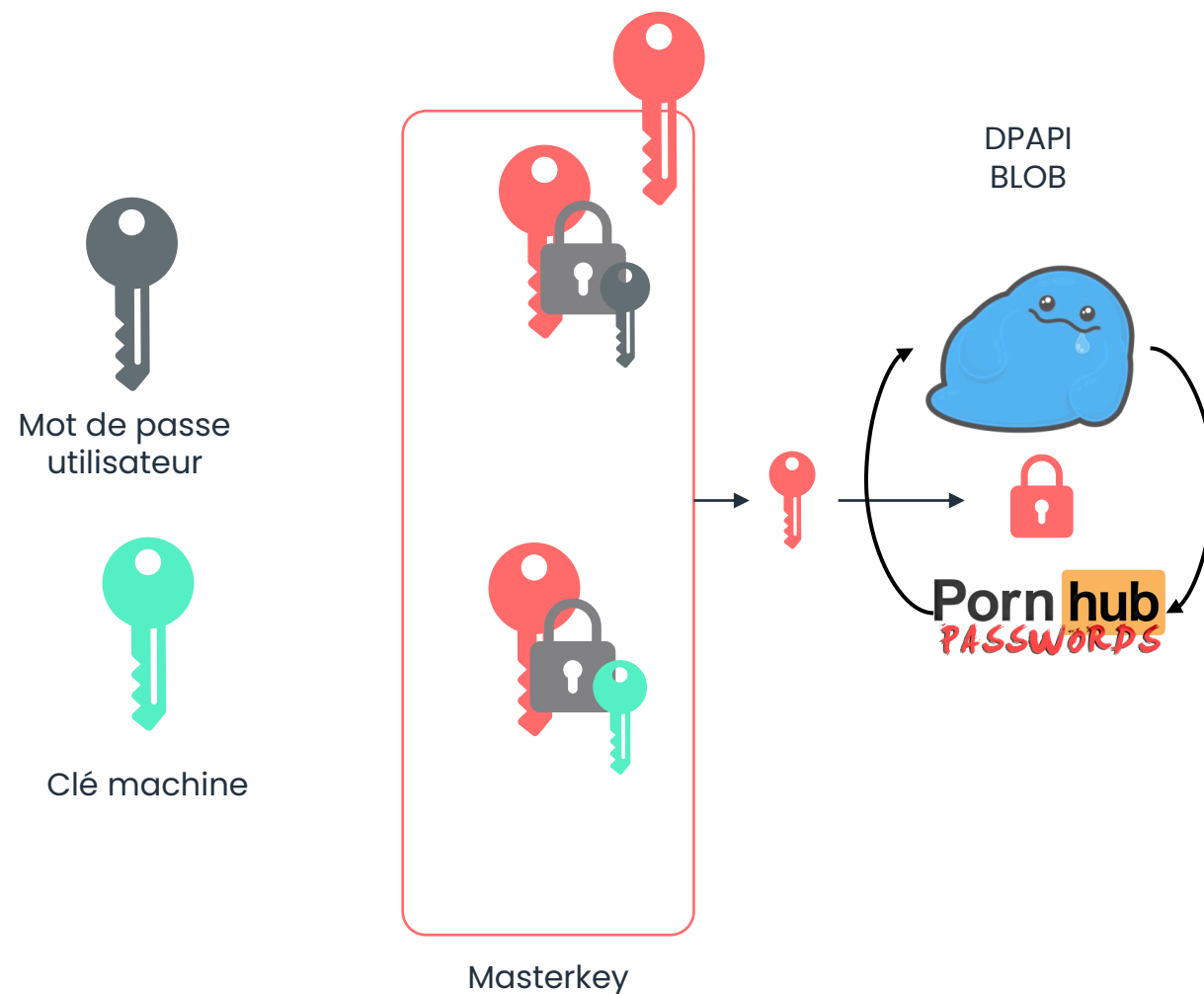
```
[BLOB]
Version      :      1 (1)
Guid Credential : DF9D8CD0-1501-11D1-8C7A-00C04FC297EB
MasterKeyVersion :      1 (1)
Guid MasterKey : 4CB85E88-4B00-440A-A52C-03C812E7F19F
Flags        : 20000000 (CRYPTPROTECT_SYSTEM)
Description   : Données d'identification locales

CryptAlgo     : 00006610 (26128) (CALG_AES_256)
Salt          : b'890b7c91493ac5741257d38d1b5b687bfeef6f57b581a4005d928526c212455b'
HMacKey       : b''
HashAlgo      : 0000800e (32782) (CALG_SHA_512)
HMac          : b'277e2ddaa44d84dd6212390e2d2fb4492eda05d4c19cd56d2cdfdbceafca4764'
Data          : b'c6132bcd1d9ddce6cac542c4c7ca84256e475e0cde23e417cdeb7e340d3e23390497
fa5ba6e1c108cd19bc8926dd15da2a6f5fdb808666e49d02915b8584ab3f813cbecee4a85079261e5c6409eb
d8c0ac8adebec3c155ef3f29ce8d2acf48019e09cf2b19498f6c0a8c38d849ae1e01b1f070159707120cc8122
2314cc5b637a0f623b2cc424543873def7de0991aa0d5ab83249d83aa43d9f372271a7070a27844f08bb801fd
529bc57f71456e912a5269b4e59e374ca5438594a143230ed75266aec50d4676fd4970f304ffcdea814ff8e23
8f1663532618edd3c6cf47902fc6aa517a40f6ef97fb66f6a79a5ba000873d3bd99332622b77046b13ea3dde
dbe41d34ce2aabcb31c565ac5fd8605392178678d414bfdc6d0a4359955916ce2b3e0a29fd72dd08ec4ef87d5
```

```
root@acherus-lehack [/data] ~> xxd blob-dpapi.txt
00000000: 0100 0000 9001 0000 0000 0000 0100 0000 .....
00000010: d08c 9ddf 0115 d111 8c7a 00c0 4fc2 97eb .....Z..0...
00000020: 0100 0000 7211 23f1 94de a844 a9ed d938 ....r.#....D...8
00000030: f2da 6ff6 0000 0020 5000 0000 4400 6f00 ..o.... P...D.o.
00000040: 6e00 6e00 e900 6500 7300 2000 6400 1920 n.n...e.s. .d..
00000050: 6900 6400 6500 6e00 7400 6900 6600 6900 i.d.e.n.t.i.f.i.
00000060: 6300 6100 7400 6900 6f00 6e00 2000 6400 c.a.t.i.o.n. .d.
```

Dissection de la DPAPI.

- La problématique :
 - Un système simple, **sans interaction** avec l'utilisateur
 - La DPAPI doit chiffrer un secret pouvant appartenir :
 - À un **utilisateur local**
 - À un **utilisateur du domaine**
 - À la **machine**
 - Les utilisateurs sont ... des utilisateurs. Ils peuvent oublier leur mot de passe
- L'utilisateur a déjà un secret qui lui est propre : son **mot de passe**. A l'ouverture de session, LSASS **déchiffre et stocke** toutes ses masterkeys avec.
 - (SID + MD4(mot de passe) pour les utilisateurs du domaine et SID + SHA1(mot de passe) pour les utilisateurs locaux)
- Windows va générer un « secret » machine, et le stocker dans LSA, la **DPAPI_SYSTEM_KEY**

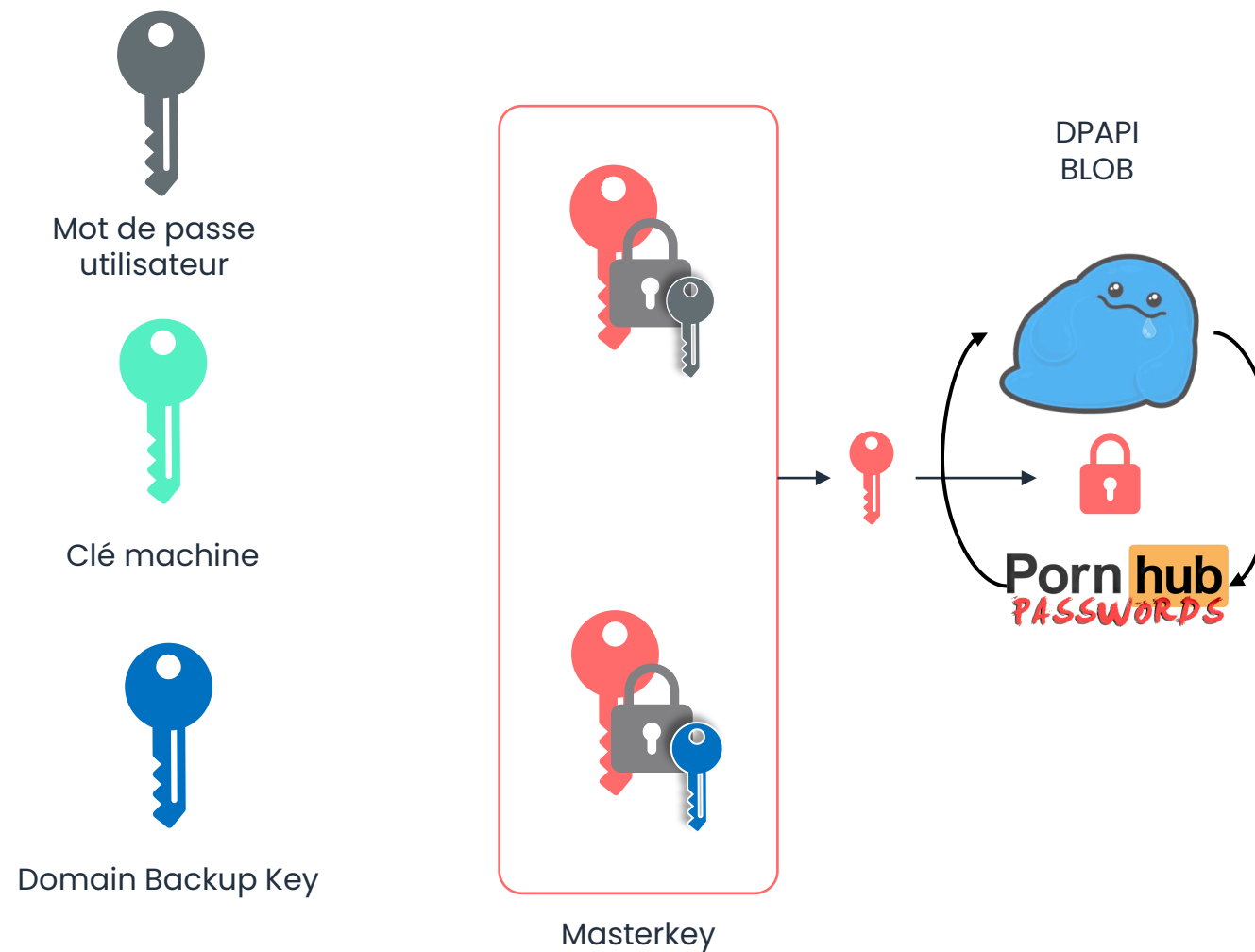


Dissection de la DPAPI.

- Quand un utilisateur oublie son mot de passe, on fait comment ?
- Fonctionne avec un système RSA de clé publique / clé privée
- Clé publique, distribuée à tous les utilisateurs du domaine
- Clé privée **unique**, stockée sur les contrôleurs de domaine
- Accessible seulement par les administrateurs du domaine ou équivalent



Dissection de la DPAPI.



```
tse@debian:/tmp/dpapi$ dpapi.py masterkey -file test/c3dd45d6-2878-4c01-8624-76f13dafe4cf
Impacket for Exegol - v0.10.1.dev1+20230318.114933.11c51f7d - Copyright 2022 Fortra - forked by ThePorgs
```

[MASTERKEYFILE]

```
Version      :          2 (2)
Guid         : c3dd45d6-2878-4c01-8624-76f13dafe4cf
Flags        :          0 (0)
Policy       :          0 (0)
MasterKeyLen: 00000088 (136)
BackupKeyLen: 00000068 (104)
CredHistLen  : 00000000 (0)
DomainKeyLen: 00000174 (372)
```

[MASTERKEY]

```
Version      :          2 (2)
Salt         : b'e37a3749053842760c863ec0c261cd2d'
Rounds       :         4650 (18000)
HashAlgo     : 00008009 (32777) (CALG_HMAC)
CryptAlgo    : 00006603 (26115) (CALG_3DES)
data         : b'948350faa3ddb4791fafd773a35620d35590cf1e7e6259d402f45fe61f74e8e6efafa502b5ec1d64048047172
ca733d740001'
```

[MASTERKEY]

```
Version      :          2 (2)
Salt         : b'3a54b9cb2f7c4d88cc87bbc76621aa14'
Rounds       :         4650 (18000)
HashAlgo     : 00008009 (32777) (CALG_HMAC)
CryptAlgo    : 00006603 (26115) (CALG_3DES)
data         : b'ead262eaffd1416613b61dc305ef5fc4a9f84a01cd7e20fe8b201992156d22f3ab9b68b274c1a6e35b758acb1'
```

[DOMAINKEY]

```
Version      :          2 (2)
Guid         : FE8411C8-69A7-4629-A4BE-92A6D21C8477
SecretLen    :          100 (256)
AccessCheckLen: 00000058 (88)
SecretData   : b'2d4551dc2dd8c0369c852fac11b514ab220adde9612a57d3441c731aa20ad260cf325380331bc0db2e9fcfe
7f21bc6342de704b8f5af8e361543269a984b79180c6d506edcba501ebfbef123cd7c67acc081140bfc5e69f93cfb0be704f1e570
4713e93f7c69e6b4688dd5a48cd90ce9a296efcf0c422ee42d507ab1869c4d26abf08d226188a94e529642ffabe9f2738021eac96
AccessCheck  : b'2191a79c4bff390e5d981f3cef4785d7662c40b5c5c653a9da6b0afbe642ef5bd9ba71d6294916ffa921a5'
```

Dissection de la DPAPI.

- La Domaine Backup Key permet de **déchiffrer tous les secrets stockés via la DPAPI** de tous les utilisateurs du domaine
- De facto, tous les administrateurs du domaine ont accès aux secrets de **tous les utilisateurs du domaine**
- De facto, tout secret stocké via la DPAPI est aussi sécurisé que l'est le compte en question, ou **les comptes administrateurs du domaine**



Dissection de la DPAPI.

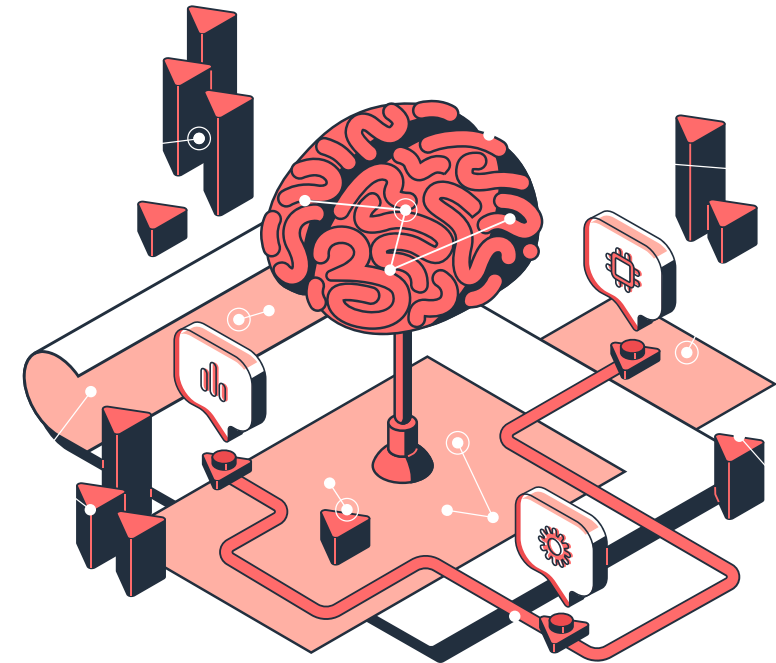
- Aucun moyen de recréer une Domain Backup Key
- En cas de compromission de la Domain Backup Key, Microsoft recommande de migrer tous les utilisateurs vers un autre domaine
- D'après un récent post de Gil Biton « **The Downfall of DPAPI Top Secret Weapon** », il est possible de régénérer la Domain Backup Key, mais cette technique n'est pas reconnue officiellement par Microsoft



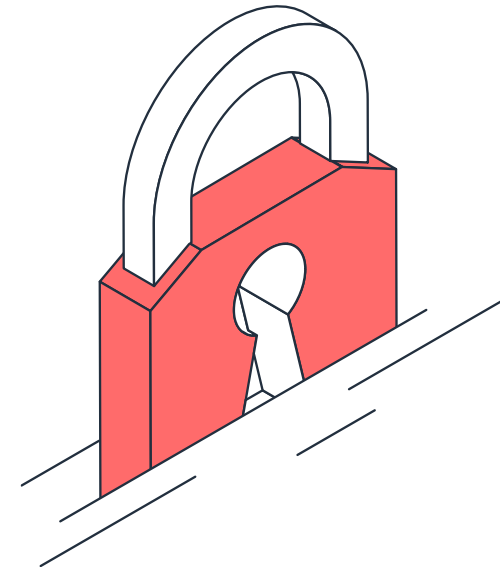
KEEP
CALM
AND
REBUILD THE
ENTIRE FOREST

Parenthèse sur DPAPI-NG.

- La DPAPI dispose de quelques limites de conceptions
 - Comment on partage un secret de manière propre ?
 - Impossible de régénérer une Domain Backup Key ?
 - Une seule clé qui permet de TOUT accéder ?
- Microsoft introduit **CNG DPAPI**, ou DPAPI-NG, sur Windows 8 & 2012 R2
- Facilite le partage de secrets entre plusieurs personnes de manière sécurisée : un secret chiffré par une machine / un utilisateur est déchiffrable sur une autre machine / un autre utilisateur (impossible avec la DPAPI)
- Permet la protection des secrets par **SID** (Utilisateurs & groupe)
- Repose sur un **clé racine KDS** (Key Distribution Services), régénérable, et il est possible d'en avoir plusieurs



- Le secret est chiffré par une **CEK** (Content Encryption Key)
- La CEK est chiffrée par une **KEK** (Key Encryption Key)
- La KEK est dérivée d'une **clé L2**, qui est elle-même générée par la **clé racine KDS**.
- La CEK chiffrée et le mot de passe chiffré sont **stockés ensemble**
- La clé L2 est obtenue via un appel à l'interface RPC **MS-GKDI** avec les bons droits.



- **Cas d'usage connus :**
 - LAPSv2
 - Exports de PFX protégés par SID
 - BitLocker
 - ASP.NET core secrets
- La DPAPI-NG ne permet le déchiffrement que via l'appel à l'interface MS-GKDI, donc nécessite un accès réseau à un contrôleur de domaine : pas viable pour des secrets nécessaires hors-accès au DC.

Parenthèse sur DPAPI-NG.

- La clé KDS est stockée dans le fichier NTDS.dit, donc **accessible aux comptes administrateurs du domaine**
- Quasi comme pour la DPAPI classique, tout repose sur **la sécurité du groupe** ayant accès au mot de passe ou **la sécurité des comptes administrateurs du domaine**

You said DPAPI-ng
will be more secure

Microsoft :

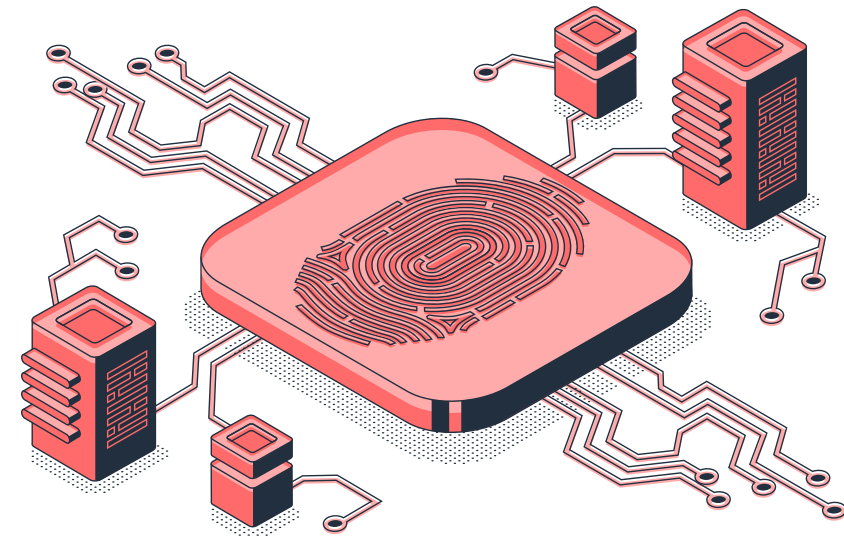


LeHack 2023

DPAPI.

Point de vue d'un
auditeur

login-securite.com





Depuis les machines et hors-ligne

- Mimikatz - <https://github.com/gentilkiwi/mimikatz>
- SharpDPAPI - <https://github.com/GhostPack/SharpDPAPI>
- Pypykatz - <https://github.com/skelsec/pypykatz>
- Lazagne - <https://github.com/AlessandroZ/LaZagne>
- Impacket via dpapi.py - <https://github.com/fortra/impacket>

Depuis le réseau

- DonPAPI - <https://github.com/login-securite/DonPAPI>
- Dploit - <https://github.com/zblurx/dploit>
- CrackMapExec (via dploit) - <https://github.com/mpgn/CrackMapExec>

Pré compromission du domaine.

- Elever ses privilèges & latéraliser
- Tous les **secrets machines** sont accessibles à partir du moment où on est administrateur local de la machine
 - Tâches planifiées
 - Certificats Machine
 - Clé Wifi



Pré compromission du domaine.

- Connection smb au share C\$
- Récupération des masterkeys machine
 - `C:\Windows\System32\Microsoft\Protect`
- Récupération des blobs DPAPI machine
 - `C:\Windows\System32\config\systemprofile\AppData\Local\Microsoft\Credentials\`
- Dump LSA pour récupérer les clefs DPAPI (machine & user)
- Déchiffrement des masterkeys via les clefs DPAPI
- Déchiffrement des blobs DPAPI avec les masterkeys déchiffrées correspondantes

Under Under the hood – Dump LSA

1. `self.__remoteOps.enableRegistry()`
2. `self.__bootKey = self.__remoteOps.getBootKey()`
3. `self.__remoteOps.saveSECURITY()`

- On peut jouer sur :
 - Le nom du fichier de sortie & son extension
 - L'accès remote au fichier (\$ADMIN != C\$/windows/System32/)
 - La temporalité entre l'accès à la bootkey et le dump du hive SECURITY

Pré compromission du domaine.

- Si le mot de passe d'un utilisateur du poste compromis est connu, alors il est possible de déchiffrer **tous ses mots de passe sur ce poste**
 - Navigateurs (mots de passe et cookies)
 - Credential Manager
 - Solutions de gestion des mots de passe et d'administration (RDG, KeePass, ect.)
- Si un utilisateur est connecté, il est possible de dump ses masterkeys déchiffrées dans **LSASS**
- Sinon, toujours possible de bruteforcer la masterkey hors-ligne avec hashcat / JtR pour retrouver le mot de passe de l'utilisateur

Pré compromission du domaine.

```
root@acherus-lehack [/data] ~> dploot credentials -u c.ponce -p capbreton -d testlab.local 192.168.56.59
```

```
1 bash
acherus-leha 1 bash*
```

Mon Jun 26 15:16

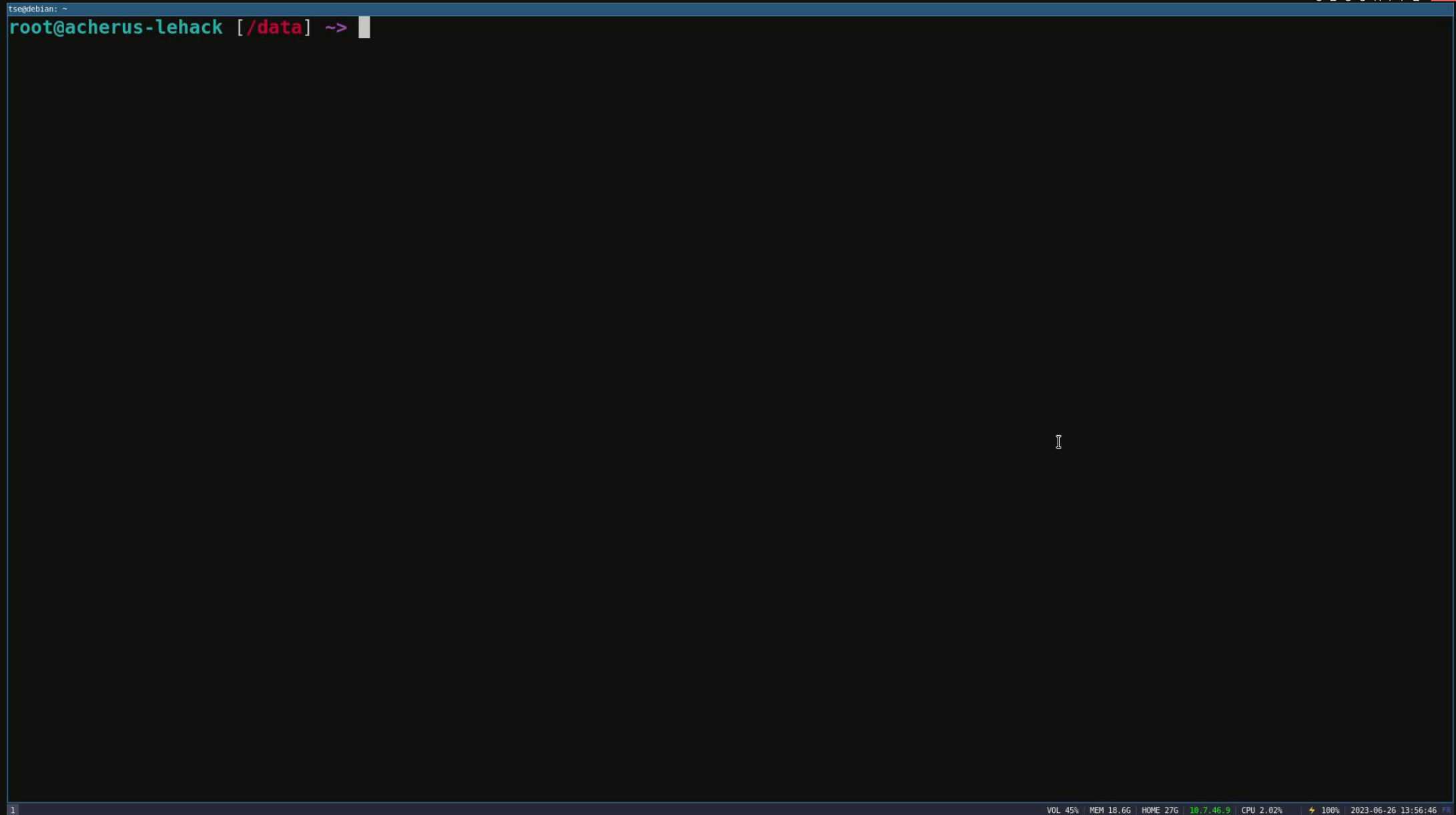
- Connection smb au share C\$
- Récupération des masterkeys de tous les utilisateurs
 - `C:\Users\c.ponce\AppData\Roaming\Microsoft\Protect`
- Récupération des blobs DPAPI des utilisateurs
 - `C:\Users\c.ponce\AppData\Local\Microsoft\Credentials`
- Déchiffrement des masterkeys via le mot de passe de l'utilisateur (saisie lors de la connexion)
- Déchiffrement des blobs DPAPI avec les masterkeys déchiffrées correspondantes

Post compromission du domaine.


- Accès à la Domain Backup Key, donc possible de **déchiffrer tous les secrets** : navigateur, credential manager, solution d'administration et gestion de mots de passe, ect.
- « **Convertir** » la compromission du domaine :
 - Compromettre le tenant Azure via les cookies sauvegardés dans les navigateurs
 - Compromettre les équipements hors domaine : cœur de réseau, NAS, solutions de backup
 - Un accès au cœur de réseau permet d'enlever le cloisonnement pour son IP
 - Compromettre d'autres domaines

Post compromission du domaine.

```
ts@debian: ~  
root@acherus-lehack [/data] ~> █
```



- Récupération de la Domain Backup Key via MS-LSAD
- Connection smb au share C\$
- Récupération des masterkeys de tous les utilisateurs
 - C:\Users\j.asselin\AppData\Roaming\Microsoft\Protect
- Récupération des données Chrome
 - C:\Users\j.asselin\AppData\Local\Google\Chrome\User Data\Local State
 - C:\Users\j.asselin\AppData\Local\Google\Chrome\User Data\Default>Login Data
 - C:\Users\j.asselin\AppData\Local\Google\Chrome\User Data\Default\Network\Cookies
- Déchiffrement des masterkeys grâce à la Domain Backup Key
- Déchiffrement des secrets Chrome avec les masterkeys déchiffrées correspondantes

Secret	Accès admin local	Accès admin local + connaissance du mot de passe utilisateur	Administrateur du domaine
Tâches planifiées	✓	✓	
Secrets navigateurs	✗	✓	
Certificats machines	✓	✓	
Certificats utilisateurs	✗	✓	
Clé Wifi	✓	✓	
Internet Explorer	✗	✓	

Et les autres secrets ?

- Beaucoup d'autres applications n'utilisent pas la DPAPI pour stocker les secrets :
 - Firefox
 - mRemoteNg
 - TightVNC (et les autres VNC)
 - LastPass
 - Putty
 - Ect.
- Généralement, récupérer les secrets en clair (si toutefois ils sont chiffrés) requiert seulement d'être administrateur local sur la machine
- DonPAPI extrait déjà tous ces secrets 🕒

Quel intérêt pour un client.

- Vision exhaustive de la gestion des secrets sur l'ensemble du parc
- Mettre en avant les mauvaises pratiques
- Plan de remédiation global sur :
 - Comptes de service
 - Comptes admin dans des tâches planifiées, lancement de services ...
 - Réutilisation de mots de passe



Quel intérêt pour un client.

DonPapi - Result for [client_name] - Mozilla Firefox

DonPapi - Result for [client_name]

file:///home/tse/Documents/Dev/test/donpapi/donpapi/14-06-2023_Client_view.html

WIFI TASKSCHEDULER CREDENTIAL-BLOB BROWSER-INTERNET_EXPLORER COOKIES SAM LSA DCC2 FILES CONNECTED-USERS LOCAL_ACCOUNT_REUSE SCOPE_AUDITED

DonPapi Audit

14/06/2023

LOGIN
SÉCURITÉ

Username	Password	Target	Type	Pillaged_from_computerid	Pillaged_from_userid
taskscheduler (25)					
LSA (325)					

Local account reuse :

Administrateur	23423aceb12312312312312313213223	SAM	DIPCM037.MYDOMAIN.NET		DIPCM037
Administrateur	23423aceb12312312312312313213223	SAM	TEPCF051.MYDOMAIN.NET		TEPCF051
Administrateur	23423aceb12312312312312313213223	SAM	BAPC0002.MYDOMAIN.NET		BAPC0002
Administrateur	23423aceb12312312312312313213223	SAM	DIPCM036.MYDOMAIN.NET		DIPCM036
Administrateur	23423aceb12312312312312313213223	SAM	DIPCM060.MYDOMAIN.NET		DIPCM060
Administrateur	23423aceb12312312312312313213223	SAM	DIPCK7244004.MYDOMAIN.NET		DIPCK7244004
Administrateur	23423aceb12312312312312313213223	SAM	DIPCM018.MYDOMAIN.NET		DIPCM018
Administrateur	23423aceb12312312312312313213223	SAM	DIPCM056.MYDOMAIN.NET		DIPCM056
Administrateur	23423aceb12312312312312313213223	SAM	HBPC0046.MYDOMAIN.NET		HBPC0046
Administrateur	23423aceb12312312312312313213223	SAM	TEPCF031.MYDOMAIN.NET		TEPCF051
Administrateur	23423aceb12312312312312313213223	SAM	DIPCM0001.MYDOMAIN.NET		DIPCM0001
Administrateur	23423aceb12312312312312313213223	SAM	CRPCM077.MYDOMAIN.NET		CRPCM077
Administrateur	23423aceb12312312312312313213223	SAM	HBPCM051.MYDOMAIN.NET		HBPCM051
Administrateur	23423aceb12312312312312313213223	SAM	DGPCF041.MYDOMAIN.NET		DGPCF041

DPAPI – Don't Put Administration Passwords In – LeHack 2023

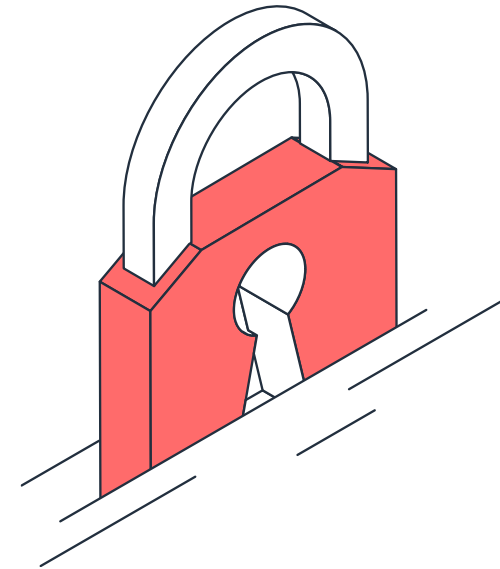
35

LeHack 2023

DPAPI.

Comment se
protéger ?

login-securite.com



Comment se protéger.

- Avoir conscience que les comptes utilisés pour **les tâches planifiées** et le lancement de services peuvent être récupérée par un attaquant avec un accès admin local a la machine
- **Ne pas stocker** de mots de passe administrateur ailleurs que dans un KeePass, et ne pas stocker le mot de passe KeePass via la DPAPI
- Mettre en place un mot de passe pour **déverrouiller** la base de mot de passe du navigateur
- Possible de **désactiver** le stockage de mots de passe pour l'authentification réseau (Credential Manager)

Types of Headaches

Migraine



Hypertension



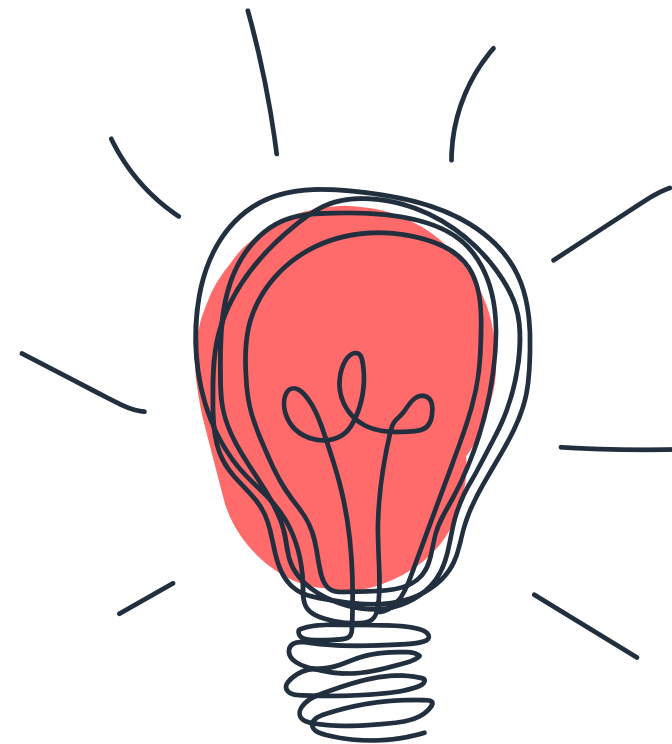
Stress



Se protéger des abus de la DPAPI



- Spoiler alert : c'est chiant
- Surveiller l'activation de **Remote Registry** et le dump de LSA Secret via un EDR (nécessaire pour récupérer les clés machines)
- Il est possible de surveiller les **accès suspects aux masterkeys** et endroits de stockage des blobs DPAPI
 - Déception : l'utilisation de **fichiers canary** pour surveiller l'accès frauduleux aux blobs de données / masterkeys detecte les actions malveillantes
- Surveiller l'accès à la **Domain Backup Key** : Event ID 4662 de type SecretObject, l'objet accédé contient "BACKUPKEY" et le masque d'accès est 0x2



- Benjamin Delpy (@gentilkiwi) pour Mimikatz et toutes les recherches sur la DPAPI
- Alberto Solino (@agsolino) pour Impacket (<https://github.com/SecureAuthCorp/impacket>). Tout ce qu'on a fait dans nos outils se base sur cette librairie
- Will Schroeder (@harmj0y) pour SharpDPAPI et plusieurs articles sur le sujet.
- Alessandro Z (@AlessandroZ) & tous les gens qui ont travaillé sur Lazagne (<https://github.com/AlessandroZ/LaZagne/wiki>)
- @Byt3bl33d3er & @mpgn_x64 pour CrackMapExec
- @Fist0urs : ma première lecture sur la DPAPI



- DonPAPI est publié dans Pypi
 - Pour l'installer : `pip install donpapi`
- Ajout du copy paste JavaScript des cookies dans le rapport
- Dump des certificats user et machines
- Tricks de bypass d'EDR pour les dump LSA

LeHack 2023

DPAPI.
des questions ?

login-securite.com