black hat USA 2019
AUGUST 3-8, 2019
MANDALAY BAY / LAS VEGAS

@paulacqure
@mjl_pl
@CQUREAcademy

https://cqureacademy.com
https://cqure.us

CQForensic: The Efficient Forensic Toolkit

Paula Januszkiewicz

CQURE: CEO, Cybersecurity Expert
CQURE Academy: Trainer
Microsoft Regional Director
MVP: Enterprise Security, MCT
paula@cqure.us

Mike Jankowski – Lorek

CQURE: Architect, Cybersecurity Expert
CQURE Academy: Trainer
Microsoft Certified Trainer
mike@cqure.pl

What does CQURE Team do?

Consulting services

→ **High quality penetration tests** with useful reports
Applications
Websites
External services (edge)
Internal services
+ configuration reviews

→ **Incident response** emergency services
– immediate reaction!

→ **Security architecture and design advisory**

→ Forensics investigation

→ Security awareness
For management and employees

Trainings

→ Security Awareness trainings for executives

→ CQURE Academy: over 40 advanced security trainings for IT Teams

→ Certificates and exams

→ Delivered all around the world only by a CQURE Team: training authors

info@cqure.us

CQURE Toolkit

Tools	
CQKawaii	CQPrefetchParser
CQUndelete	CQRDCache
CQWSLMon	CQSecretsDumper
CQEVTXRecovery	CQPSWinRMHistory
CQRDCManDecrypter	CQDPAPINGPFXDecrypter
CQDPAPIKeePassDBDecryptor	CQHashDumpv2
CQMasterKeyAD	CQSysmonNetAnalyzer
CQRegKeyLastWriteTime	CQReflectivePELoader

Tools	
CQMasterKeyAD	DPAPIBlobCreator
CQDPAPIKeePassDBDecryptor	DPAPINGDecrypter
CQDPAPIEncDec	CQAspNetCoreDecryptData.
CQDPAPIExportPFXFromAD	CQAspNetCoreMasterKeyCreate
CQRDCManDecrypter	CQAspNetCoreEncryptData
CQDPAPINGPFXDecrypter	
CQDPAPINGDNCoreMasterKeyDecrypter	*CQImpersonateWithSeTcb

Classic Data Protection API

- Based on the following components:

  Password, data blob, entropy

- Is not prone to password resets!

  Protects from outsiders when being in offline access

  Effectively protects users data

- Stores the password history

  You need to be able to get access to some of your passwords from the past

Conclusion: OS greatly helps us to protect secrets

Getting the: Classic DPAPI Secrets

DPAPI (classic)

A. MasterKey

1. pwdhash = MD4(password) or SHA1(password)
2. pwdhash_key = HMACSHA1(pwdhash, user_sid)
3. PBKDF2(…, pwdhash_key,…), another elements from the file. Windows 10 no domain: SHA512, AES-256, 8000 rounds
4. Control - HMACSHA512

B. CREDHIST

1. pwdhash = MD4(password) or SHA1(password)
2. pwdhash_key = HMACSHA1(pwdhash, user_sid)
3. PBKDF2(…, pwdhash_key,…), another elements from the file. Windows 10 no domain: SHA512, AES-256, 8000 rounds
4. Control - HMACSHA512

C. DPAPI blob Algorithms are written in the blob itself.

Classic DPAPI Flow: getting the system's secrets (easy)

DPAPI + AD

CLIENT

Local LSASS process (local masterkey can't be decrypted)

DPAPI

CryptUnprotectData()

```
01 00 00 00  d0 8c 9d df  01 15 d1 11  8c 7a 00 c0
4f c2 97 eb  01 00 00 00  ......
```

DPAPI-PROTECTED BLOB

RPC Call

BackupKey(masterkey)

Decrypted masterkey

Scenario: offline changed user password or local masterkey can't be decrypted

AD SERVER

LSASS process

G$BCKUPKEY_PREFERRED

G$BCKUPKEY_940db612-ee8f-4a31-84b3-8f80c25be855

Cached Logons

Windows Vista / 2008 +

The encryption algorithm is AES128.

The hash is used to verify authentication is calculated as follows:

```
MSDCC2 = PBKDF2(HMAC-SHA1, Iterations,
DCC1, LowerUnicode(username))
```

with DCC 1 calculated in the same way as for 2003 / XP.

Usage in the attack

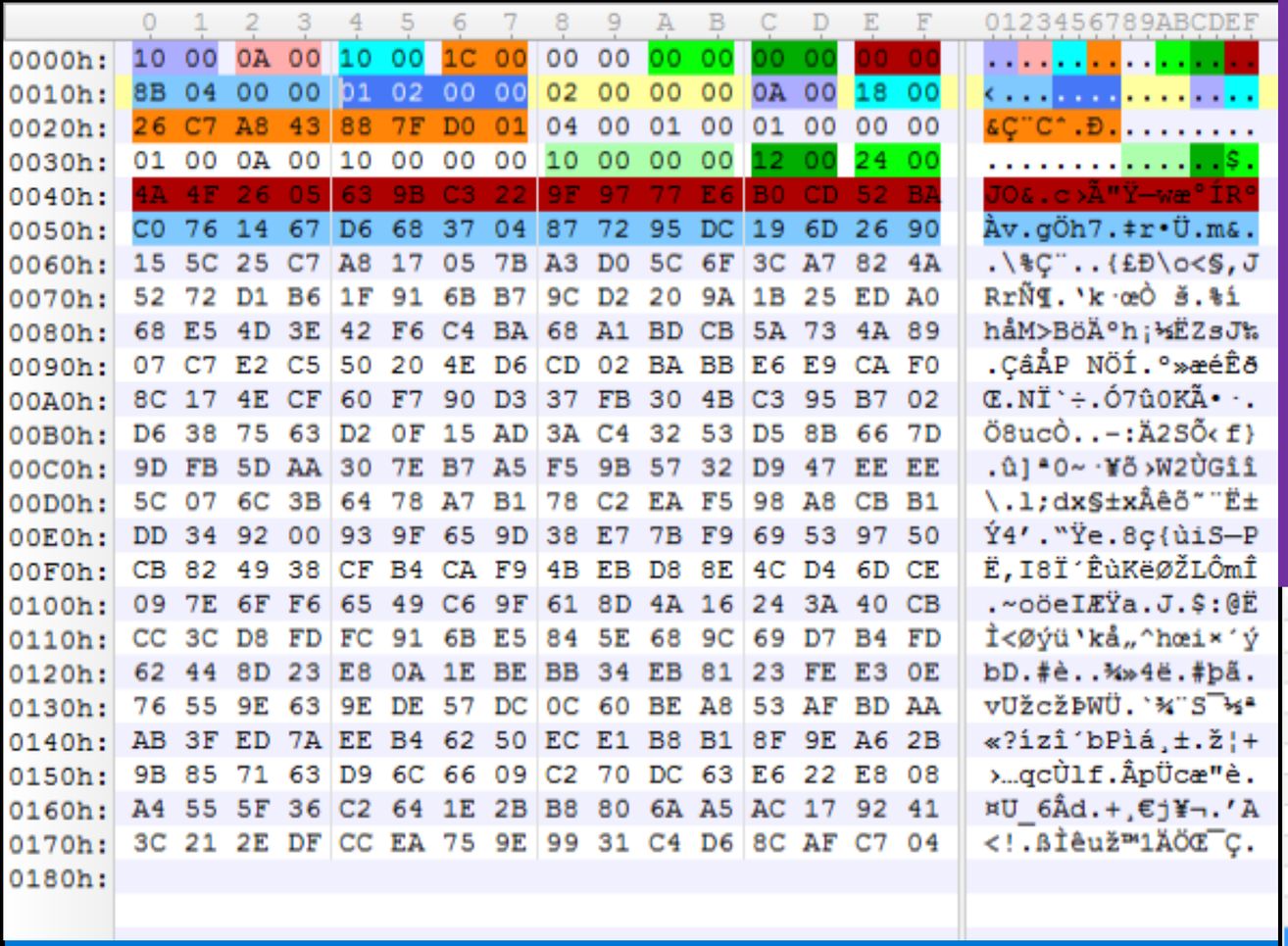
There is actually not much of a difference with XP / 2003!
No additional salting.

PBKDF2 introduced a new variable: the number of iterations SHA1 with the same salt as before (username).

Getting the: cached data

MSDCC2

1. bootkey: classes from HKLM\SYSTEM\CCS\Control\Lsa + [class names for: Data, GBG, JD, Skew1] (+arrays' permutations)
   int[] permutationBootKey = new int[] { 0x8, 0x5, 0x4, 0x2, 0xb, 0x9, 0xd, 0x3, 0x0, 0x6, 0x1, 0xc, 0xe, 0xa, 0xf, 0x7 };
2. PolEKList: HKLM\SECURITY\Policy\PolEKList [default value]
3. lsakey: AES_DECRYPT(key, data) -> AES(bootkey, PolEKList)
4. NL$KM secret: HKLM\SECURITY\Policy\Secrets\NL$KM
5. nlkm_decrypted: AES_DECRYPT(lsakey, NL$KM secret)
6. Cache_Entry{id} -> HKLM\SECURITY\Cache\NL${id}
7. cache_entry_decrypted -> AES_DECRYPT(nlkm_decrypted, Cache_Entry{id})



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	10	00	0A	00	10	00	1C	00	00	00	00	00	00	00	00	00	.....
0010h:	8B	04	00	00	01	02	00	00	02	00	00	00	0A	00	18	00	‹...
0020h:	26	C7	A8	43	88	7F	D0	01	04	00	01	00	01	00	00	00	&Ç¨C^.Ð.....
0030h:	01	00	0A	00	10	00	00	00	10	00	00	00	12	00	24	00	.....$.
0040h:	4A	4F	26	05	63	9B	C3	22	9F	97	77	E6	B0	CD	52	BA	JO&.c›Ã"Ÿ—wæ°ÍR°
0050h:	C0	76	14	67	D6	68	37	04	87	72	95	DC	19	6D	26	90	Àv.gÖh7.‡r•Ü.m&.
0060h:	15	5C	25	C7	A8	17	05	7B	A3	D0	5C	6F	3C	A7	82	4A	.\%Ç¨..{£Ð\o<§,J
0070h:	52	72	D1	B6	1F	91	6B	B7	9C	D2	20	9A	1B	25	ED	A0	RrÑ¶.'k·œÒ š.%í
0080h:	68	E5	4D	3E	42	F6	C4	BA	68	A1	BD	CB	5A	73	4A	89	håM>BöÄºh¡½ËZsJ‰
0090h:	07	C7	E2	C5	50	20	4E	D6	CD	02	BA	BB	E6	E9	CA	F0	.ÇâÅP NÖÍ.º»æéÊð
00A0h:	8C	17	4E	CF	60	F7	90	D3	37	FB	30	4B	C3	95	B7	02	Œ.NÏ`÷.Ó7û0KÃ•·.
00B0h:	D6	38	75	63	D2	0F	15	AD	3A	C4	32	53	D5	8B	66	7D	Ö8ucÒ..-:Ä2SÕ‹f}
00C0h:	9D	FB	5D	AA	30	7E	B7	A5	F5	9B	57	32	D9	47	EE	EE	.û]ª0~·¥õ›W2ÙGîî
00D0h:	5C	07	6C	3B	64	78	A7	B1	78	C2	EA	F5	98	A8	CB	B1	\.l;dx§±xÂêõ˜¨Ë±
00E0h:	DD	34	92	00	93	9F	65	9D	38	E7	7B	F9	69	53	97	50	Ý4'."Ÿe.8ç{ùiS—P
00F0h:	CB	82	49	38	CF	B4	CA	F9	4B	EB	D8	8E	4C	D4	6D	CE	Ë‚I8Ï´ÊùKëØŽLÔmÎ
0100h:	09	7E	6F	F6	65	49	C6	9F	61	8D	4A	16	24	3A	40	CB	.~oöeIÆŸa.J.$:@Ë
0110h:	CC	3C	D8	FD	FC	91	6B	E5	84	5E	68	9C	69	D7	B4	FD	Ì<Øýü'kå„^hœi×´ý
0120h:	62	44	8D	23	E8	0A	1E	BE	BB	34	EB	81	23	FE	E3	0E	bD.#è..¾»4ë.#þã.
0130h:	76	55	9E	63	9E	DE	57	DC	0C	60	BE	A8	53	AF	BD	AA	vUžcžÞWÜ.`¾¨S¯½ª
0140h:	AB	3F	ED	7A	EE	B4	62	50	EC	E1	B8	B1	8F	9E	A6	2B	«?ízî´bPìá¸±.ž¦+
0150h:	9B	85	71	63	D9	6C	66	09	C2	70	DC	63	E6	22	E8	08	›…qcÙlf.ÂpÜcæ"è.
0160h:	A4	55	5F	36	C2	64	1E	2B	B8	80	6A	A5	AC	17	92	41	¤U_6Âd.+¸€j¥¬.'A
0170h:	3C	21	2E	DF	CC	EA	75	9E	99	31	C4	D6	8C	AF	C7	04	<!.ßÌêuž™1ÄÖŒ¯Ç.
0180h:																	

Encrypted Cached Credentials
DK = PBKDF2(PRF, Password, Salt, c, dkLen)

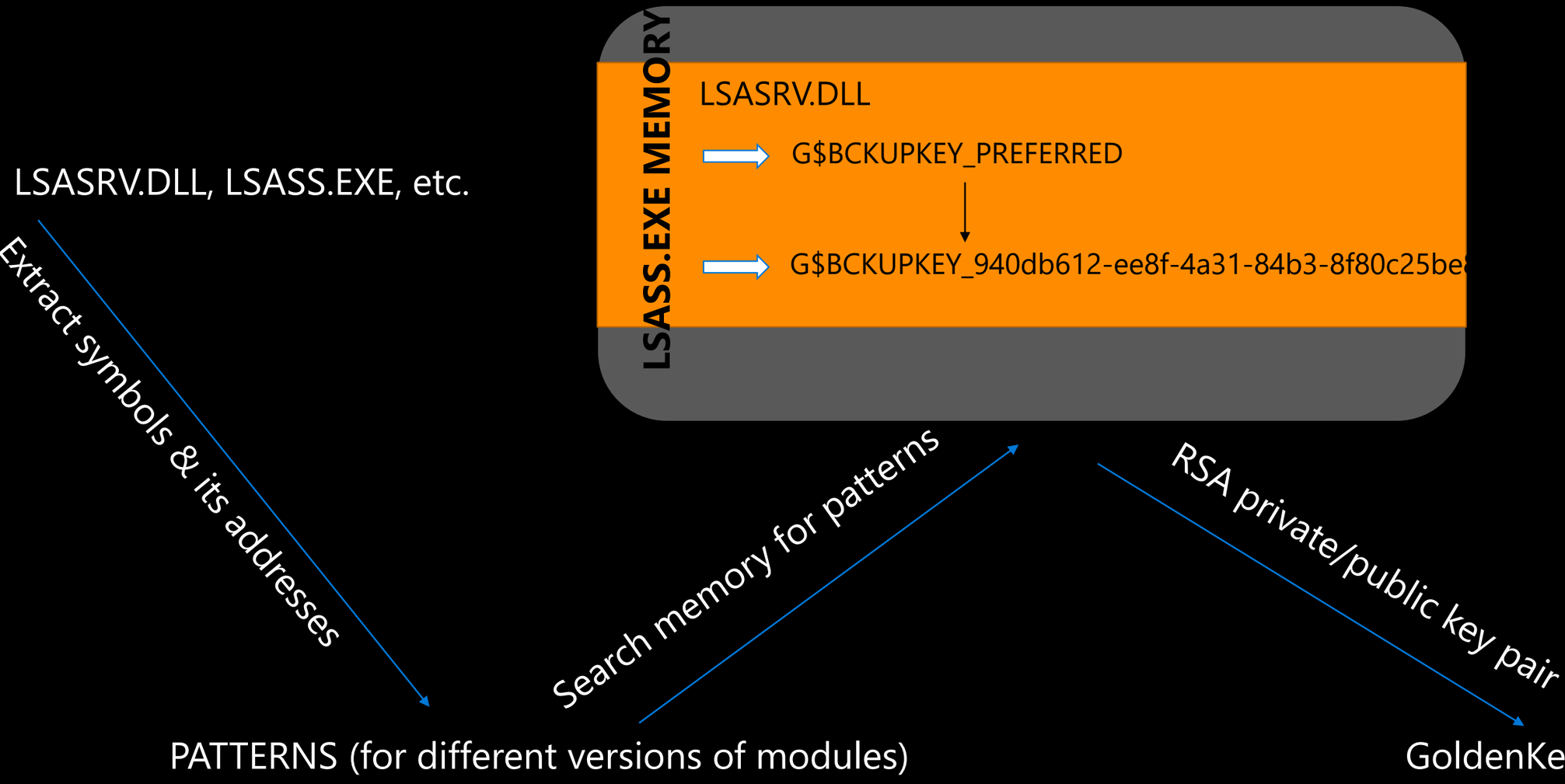
Microsoft's implementation: MSDCC2=
PBKDF2(HMAC-SHA1, DCC1, username, 10240, 16)

Encrypted Cached Credentials: Legend

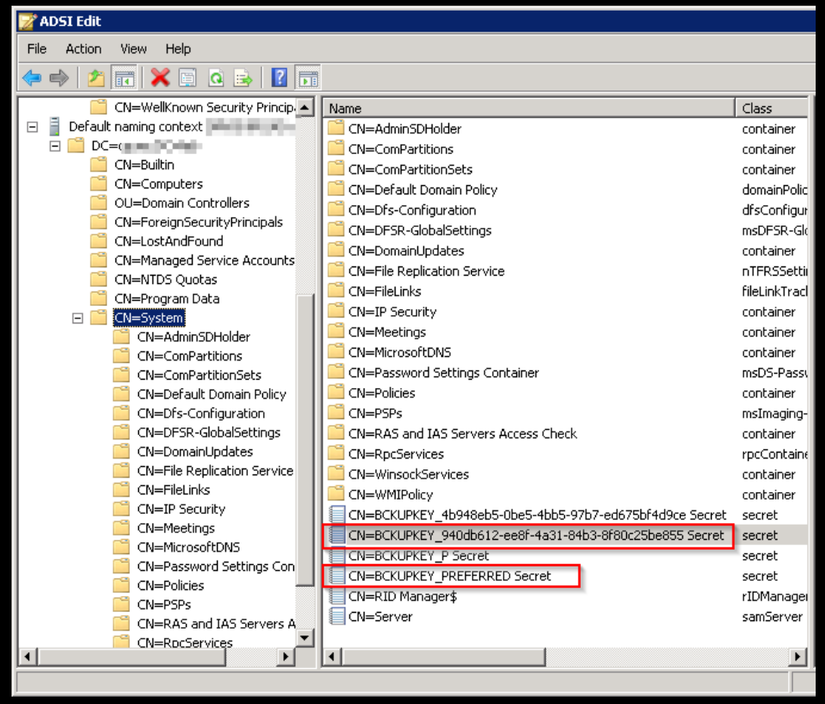
Name	Value	Start	Size	Color	Comment
struct Header h		0h	96	Fg: Bg:	
ushort uname_len	16	0h	2	Fg: Bg:	
ushort domain_len	10	2h	2	Fg: Bg:	
ushort mail_nick_len	16	4h	2	Fg: Bg:	
ushort cn_len	28	6h	2	Fg: Bg:	
ushort u1	0	8h	2	Fg: Bg:	
ushort logon_script_len	0	Ah	2	Fg: Bg:	
ushort profile_path_len	0	Ch	2	Fg: Bg:	
ushort home_dir_len	0	Eh	2	Fg: Bg:	
uint user_sid	1163	10h	4	Fg: Bg:	
uint primary_group_id	513	14h	4	Fg: Bg:	
uint u2	2	18h	4	Fg: Bg:	
ushort group_sids_len	10	1Ch	2	Fg: Bg:	
ushort domain_netbios_name...	24	1Eh	2	Fg: Bg:	
FILETIME last_local_logon	04/25/2015 18:47:22	20h	8	Fg: Bg:	
ushort u3	4	28h	2	Fg: Bg:	
ushort u4	1	2Ah	2	Fg: Bg:	
uint u5	1	2Ch	4	Fg: Bg:	
ushort u6	1	30h	2	Fg: Bg:	
ushort u7	10	32h	2	Fg: Bg:	
uint u8	16	34h	4	Fg: Bg:	
uint u9	16	38h	4	Fg: Bg:	
ushort domain_name_len	18	3Ch	2	Fg: Bg:	
ushort email_len	36	3Eh	2	Fg: Bg:	
byte iv[16]	JO&|c›Ã"Ÿ—wæ°ÍR°	40h	16	Fg: Bg:	
byte cksum[16]	Àv¶gÖh7┘‡r•Ü ├m&□	50h	16	Fg: Bg:	

Classic DPAPI Flow: getting the user's secrets

Retrieving Golden Key from LSA – Mimikatz' way



Retrieving Golden Key from LSA – CQURE's way



AD secret? HOW?!

? ? ? ? ? ? ? ?

LSASS.EXE MEMORY

LSASRV.DLL

G$BCKUPKEY_PREFERRED

G$BCKUPKEY_940db612-ee8f-4a31-84b3-8f80c25be

LsaRetrievePrivateData()

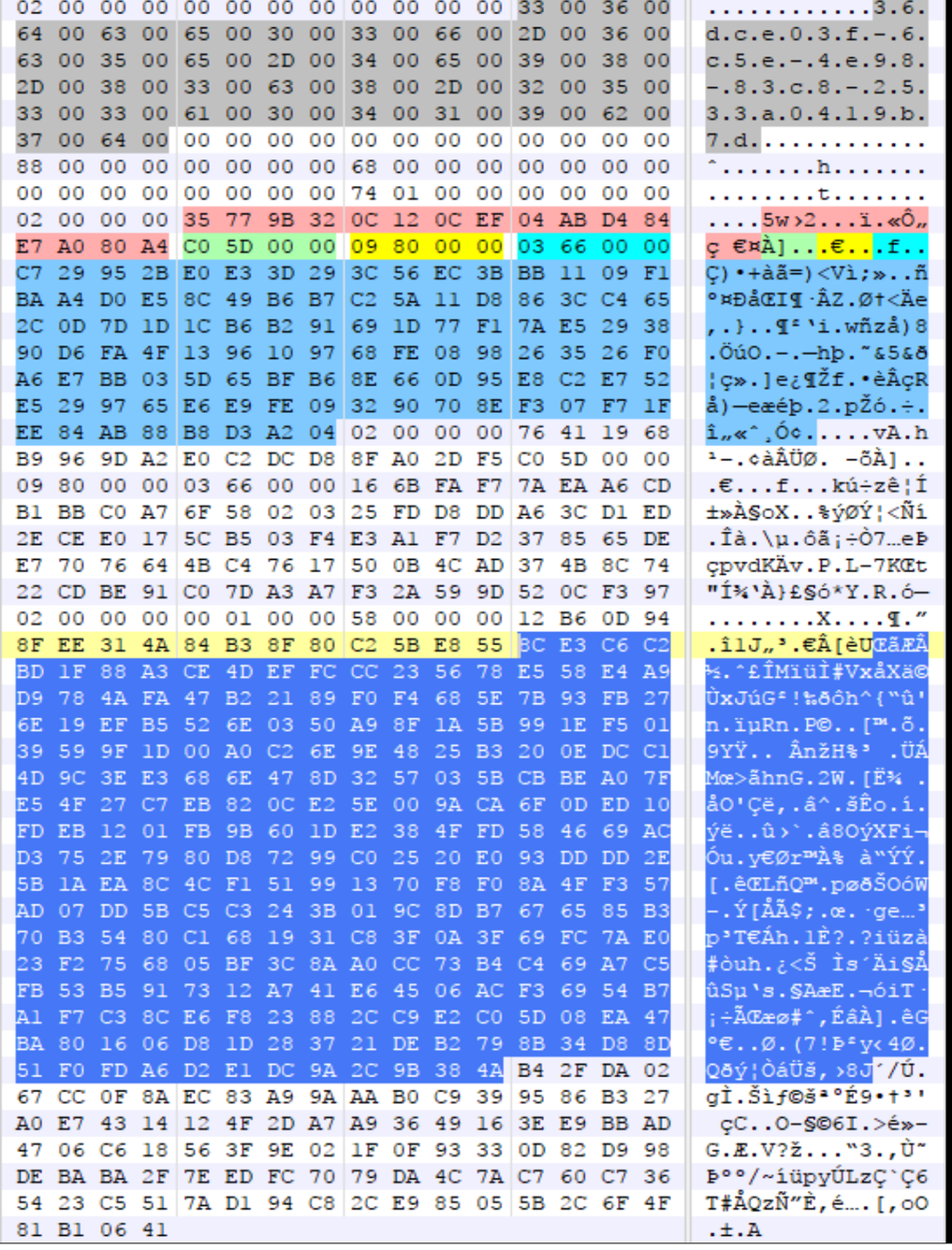
RSA private/public key pair

GoldenKey.pfx

DPAPI-AD: How (the hell) did we do it?

Dude, look in the AD...

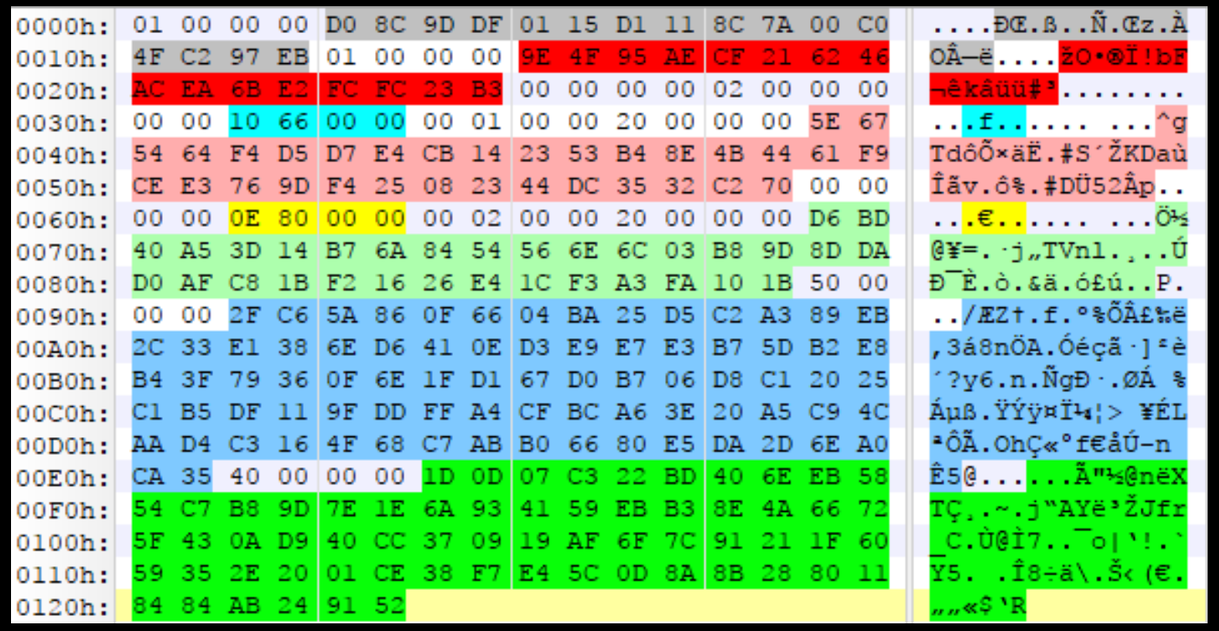
DomainKey contains some GUID and 256-byte len secret – RSA??



Name	Value	Start				Comment
struct MasterKeyFile mkf		0h				
uint version	2	0h				
uint unknown1	0	4h				
uint unknown2	0	8h	4h			
wchar_t guid[36]	36dce03f-6c5e-4e98-83c8-2533a0419b7d	Ch	48h			
uint unknown3	0	54h	4h			
uint unknown4	0	58h	4h			
uint policy	0	5Ch	4h			
quad masterkeyLen	136	60h	8h	Fg:	Bg:	
quad backupkeyLen	104	68h	8h	Fg:	Bg:	
quad credhistLen	0	70h	8h	Fg:	Bg:	
quad domainkeyLen	372	78h	8h	Fg:	Bg:	
struct MasterKey masterkey		80h	88h	Fg:	Bg:	
uint version	2	80h	4h	Fg:	Bg:	
byte iv[16]	5w›2□□□ï□«Ô„ç €¤	84h	10h	Fg:	Bg:	
uint rounds	24000	94h	4h	Fg:	Bg:	
uint hashAlgo	32777	98h	4h	Fg:	Bg:	
uint cipherAlgo	26115	9Ch	4h	Fg:	Bg:	
byte cipherText[104]	Ç)•+àã=)<Vì;»□ ñº¤ÐåŒI¶·ÂZ□Ø†<Ä…	A0h	68h	Fg:	Bg:	
struct MasterKey backupkey		108h	68h	Fg:	Bg:	
struct DomainKey domainkey		170h	174h	Fg:	Bg:	
uint version	2	170h	4h	Fg:	Bg:	
uint secretLen	256	174h	4h	Fg:	Bg:	
uint accesscheckLen	88	178h	4h	Fg:	Bg:	
struct GUID guidKey	940db612-ee8f-4a31-84b3-8f80c25be855	17Ch	10h	Fg:	Bg:	
byte encryptedSecret[256]	ŒãÆÄ½□ˆ£ÍMïüÍ#VxåXä©ÚxJúG²!‰ðõ…	18Ch	100h	Fg:	Bg:	
byte accessCheck[88]	´/Ú□gÌ□Šìƒ©šª°É9•†³' çC□□O-§©6I□…	28Ch	58h	Fg:	Bg:	

Demo:
What about KeePass?

DPAPI in pictures
Example: KeePass ProtectedUserKey.bin



0000h:	01	00	00	00	D0	8C	9D	DF	01	15	D1	11	8C	7A	00	C0	....ÐŒ.ß..Ñ.Œz.À
0010h:	4F	C2	97	EB	01	00	00	00	9E	4F	95	AE	CF	21	62	46	OÂ—ë....žO•®Ï!bF
0020h:	AC	EA	6B	E2	FC	FC	23	B3	00	00	00	00	02	00	00	00	¬êkâüü#³........
0030h:	00	00	10	66	00	00	00	01	00	00	20	00	00	00	5E	67	...f.... ....^g
0040h:	54	64	F4	D5	D7	E4	CB	14	23	53	B4	8E	4B	44	61	F9	TdôÕ×äË.#S´ŽKDaù
0050h:	CE	E3	76	9D	F4	25	08	23	44	DC	35	32	C2	70	00	00	Îãv.ô%.#DÜ52Âp..
0060h:	00	00	0E	80	00	00	00	02	00	00	20	00	00	00	D6	BD	..€..... ....Ö½
0070h:	40	A5	3D	14	B7	6A	84	54	56	6E	6C	03	B8	9D	8D	DA	@¥=.·j„TVnl.¸..Ú
0080h:	D0	AF	C8	1B	F2	16	26	E4	1C	F3	A3	FA	10	1B	50	00	Ð¯È.ò.&ä.ó£ú..P.
0090h:	00	00	2F	C6	5A	86	0F	66	04	BA	25	D5	C2	A3	89	EB	../ÆZ†.f.º%ÕÂ£‰ë
00A0h:	2C	33	E1	38	6E	D6	41	0E	D3	E9	E7	E3	B7	5D	B2	E8	,3á8nÖA.Óéçã·]²è
00B0h:	B4	3F	79	36	0F	6E	1F	D1	67	D0	B7	06	D8	C1	20	25	´?y6.n.ÑgÐ·.ØÁ %
00C0h:	C1	B5	DF	11	9F	DD	FF	A4	CF	BC	A6	3E	20	A5	C9	4C	Áµß.ŸÝÿ¤Ï¼¦> ¥ÉL
00D0h:	AA	D4	C3	16	4F	68	C7	AB	B0	66	80	E5	DA	2D	6E	A0	ªÔÃ.OhÇ«°f€åÚ-n
00E0h:	CA	35	40	00	00	00	1D	0D	07	C3	22	BD	40	6E	EB	58	Ê5@.....Ã"½@nëX
00F0h:	54	C7	B8	9D	7E	1E	6A	93	41	59	EB	B3	8E	4A	66	72	TÇ¸.~.j"AYë³ŽJfr
0100h:	5F	43	0A	D9	40	CC	37	09	19	AF	6F	7C	91	21	1F	60	_C.Ù@Ì7..¯o|'!.`
0110h:	59	35	2E	20	01	CE	38	F7	E4	5C	0D	8A	8B	28	80	11	Y5. .Î8÷ä\.Š‹(€.
0120h:	84	84	AB	24	91	52											„„«$'R

Name	Value	Start	Size	Color	Comment
struct DPAPIBlob blob		0h	126h	Fg: Bg:	
uint version	1	0h	4h	Fg: Bg:	
struct GUID provider	df9d8cd0-1501-11d1-8c7a-00c04fc297eb	4h	10h	Fg: Bg:	
uint mkversion	1	14h	4h	Fg: Bg:	
struct GUID mkguid	ae954f9e-21cf-4662-acea-6be2fcfc23b3	18h	10h	Fg: Bg:	
uint flags	0	28h	4h	Fg: Bg:	
uint descriptionLen	2	2Ch	4h	Fg: Bg:	
wstring description[1]		30h	2h	Fg: Bg:	
uint cipherAlgo	26128	32h	4h	Fg: Bg:	
uint keyLen	256	36h	4h	Fg: Bg:	
uint saltLen	32	3Ah	4h	Fg: Bg:	
byte salt[32]	^gTdôÕ×äË□#S´ŽKDaùÎãv□ô%□#DÜ5...	3Eh	20h	Fg: Bg:	
uint strongLen	0	5Eh	4h	Fg: Bg:	
uint hashAlgo	32782	62h	4h	Fg: Bg:	
uint hashLen	512	66h	4h	Fg: Bg:	
uint hmacLen	32	6Ah	4h	Fg: Bg:	
byte hmac[32]	Ö½@¥=□·j„TVnl□¸□□ÚÐ¯È□ò□&ä□ó...	6Eh	20h	Fg: Bg:	
uint cipherTextLen	80	8Eh	4h	Fg: Bg:	
byte cipherText[80]	/ÆZ†□f□º%ÕÂ£‰ë,3á8nÖA□Óéçã·]²è...	92h	50h	Fg: Bg:	
uint signLen	64	E2h	4h	Fg: Bg:	
byte sign[64]	□ □Ã"½@nëXTÇ¸□~□j"AYë³ŽJfr_C Ù...	E6h	40h	Fg: Bg:	

The master password for KeePass files encrypted & stored as cipherText (80 bytes)

DPAPI blob:
Legend

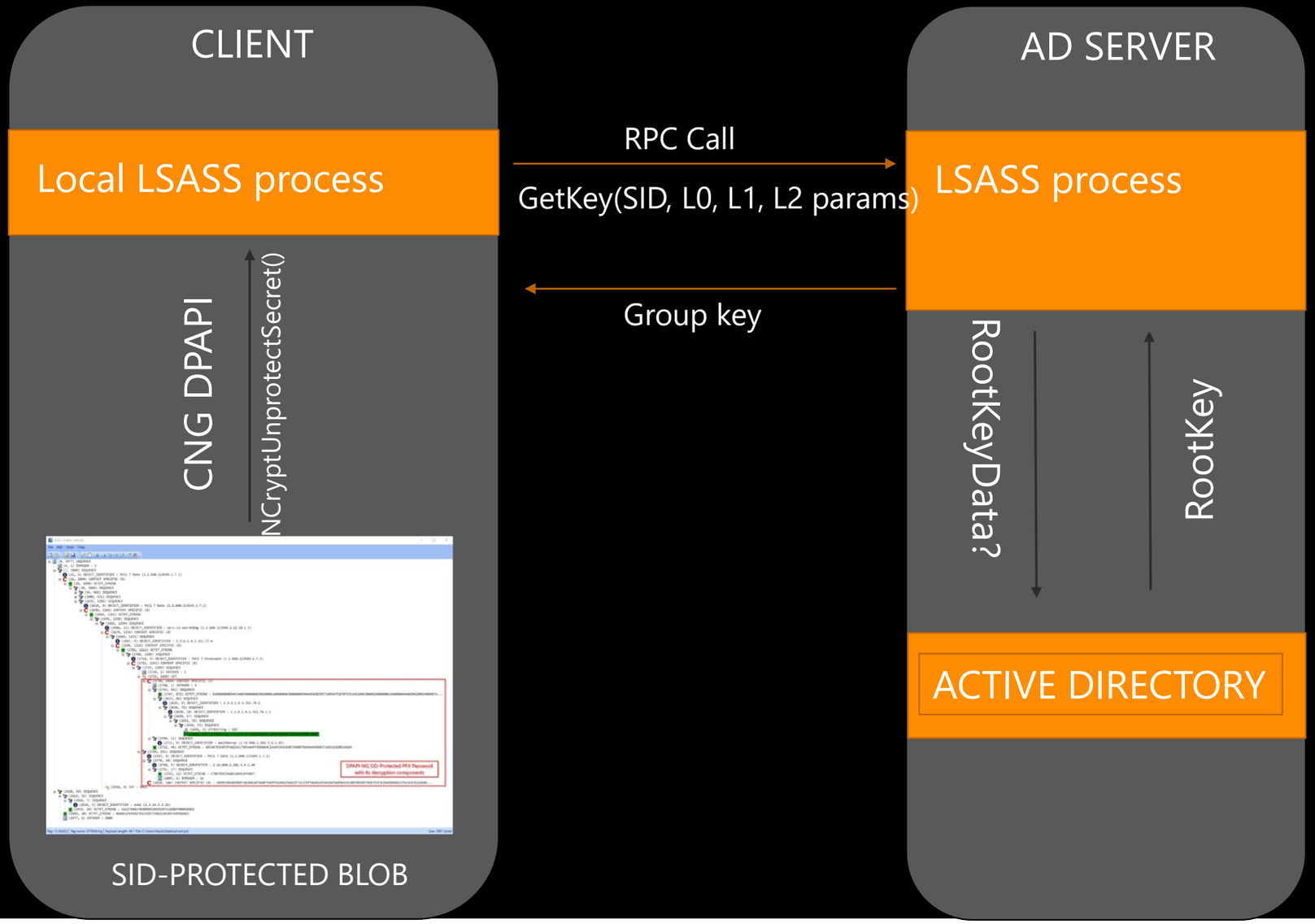
Demo:
What about RDP Connections?

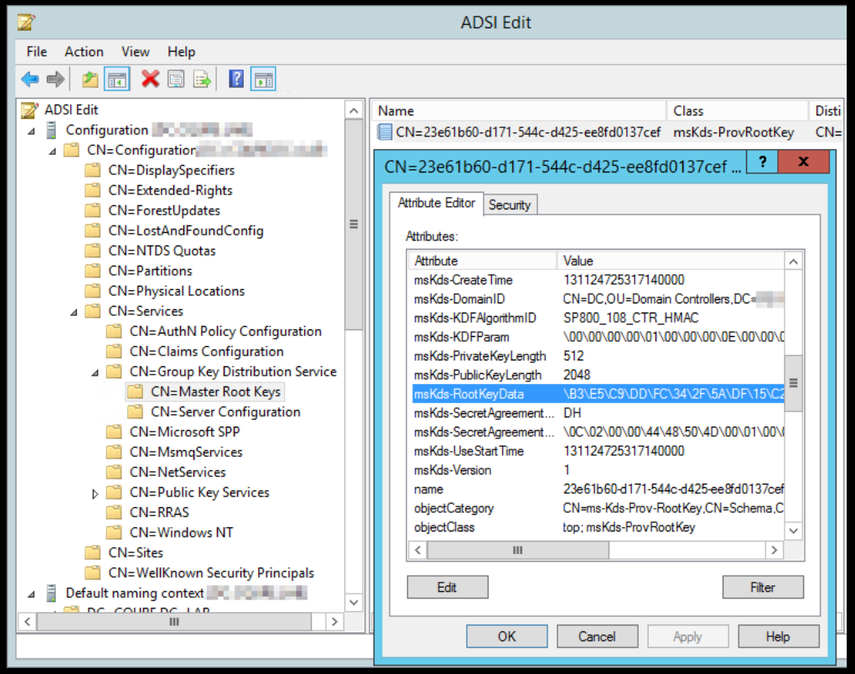
Getting the: DPAPI-NG Secrets

DPAPI-NG

A. RootKey Algorithms Key derivation function:
SP800_108_CTR_HMAC (SHA512) Secret agreement: Diffie-Hellman
B. DPAPI blob Key derivation: KDF_SP80056A_CONCAT
After getting the key, there is a need for decryption: Key wrap algorithm: RFC3394 (KEK -> CEK) Decryption: AES-256-GCM (CEK, Blob)

DPAPI-NG: Data encryption flow





DPAPI-NG: Protected data encoded as ASN.1 blob

- KEK (Key Encryption Key) stored as DPAPI blob
- Forced by protection descriptor LOCAL=user
- Key Wrap (RFC3394) contains encrypted CEK (Content Encryption Key)
- Data encrypted by CEK

DPAPI-NG: getting to SID-Protected PFX files

https://cqu.re/forensictoolkit

black hat USA 2019

AUGUST 3–8, 2019

MANDALAY BAY / LAS VEGAS

@paulacqure
@mjl_pl
@CQUREAcademy

https://cqureacademy.com
https://cqure.us

Paula Januszkiewicz

CQURE: CEO, Cybersecurity Expert
CQURE Academy: Trainer
Microsoft Regional Director
MVP: Enterprise Security, MCT
paula@cqure.us

Mike Jankowski – Lorek

CQURE: Architect, Cybersecurity Expert
CQURE Academy: Trainer
Microsoft Certified Trainer
mike@cqure.pl

Thank you!

#BHUSA @BLACK HAT EVENTS