le cnam

SEC 102 - TP07
Analyse statique

Date	Version	Description	Auteurs
16/04/2023	1	Retranscription collective des informations présentées en cours	Bartel Cantin Bialas Alexis Dufourt Marvin Faussurier Marc N'Guessan Jérémy

Sommaire

A. Analyse de procexp.exe

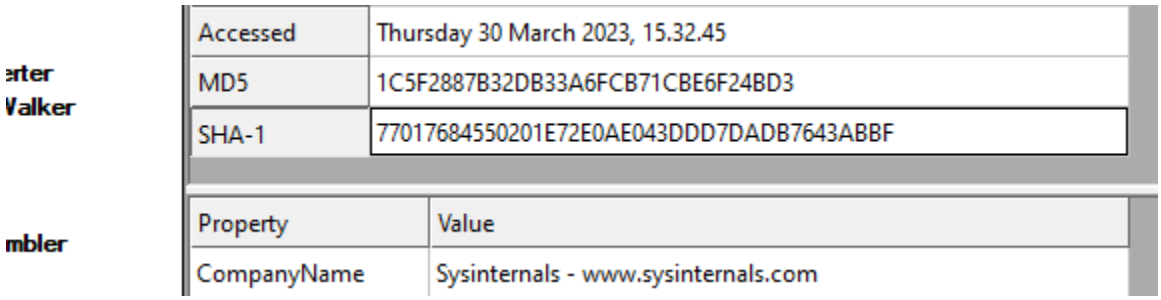
1. Information sur le fichier

Après avoir téléchargé la suite d'outils Sysinternals nous pouvons avoir accès aux informations suivante :

Nom du fichier :	procexp.exe
Date de compilation (time stamp) :	Thursday, November 10, 2022 7:21:28 PM
Taille du fichier (Size of Image) :	1020416

2. Signature du hash

La signature du hash d'un fichier est une empreinte numérique unique qui permet d'identifier et de vérifier l'intégrité d'un fichier.

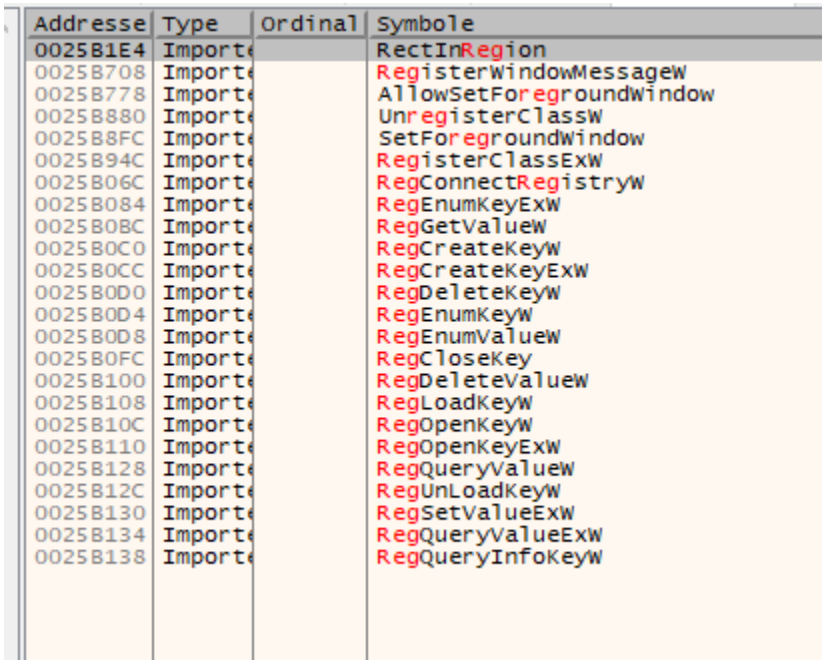



```
→  SysinternalsSuite md5 procexp.exe
MD5 (procexp.exe) = 1c5f2887b32db33a6fcb71cbe6f24bd3
```

```
→  SysinternalsSuite shasum -a 256  procexp64.exe
2cd8fa82ffccf17a8a20178bc7145ea40b837c37f7383e5b15a2243cc601dd58  procexp64.exe
→  SysinternalsSuite
```

3. Interaction de procexp.exe avec le SE

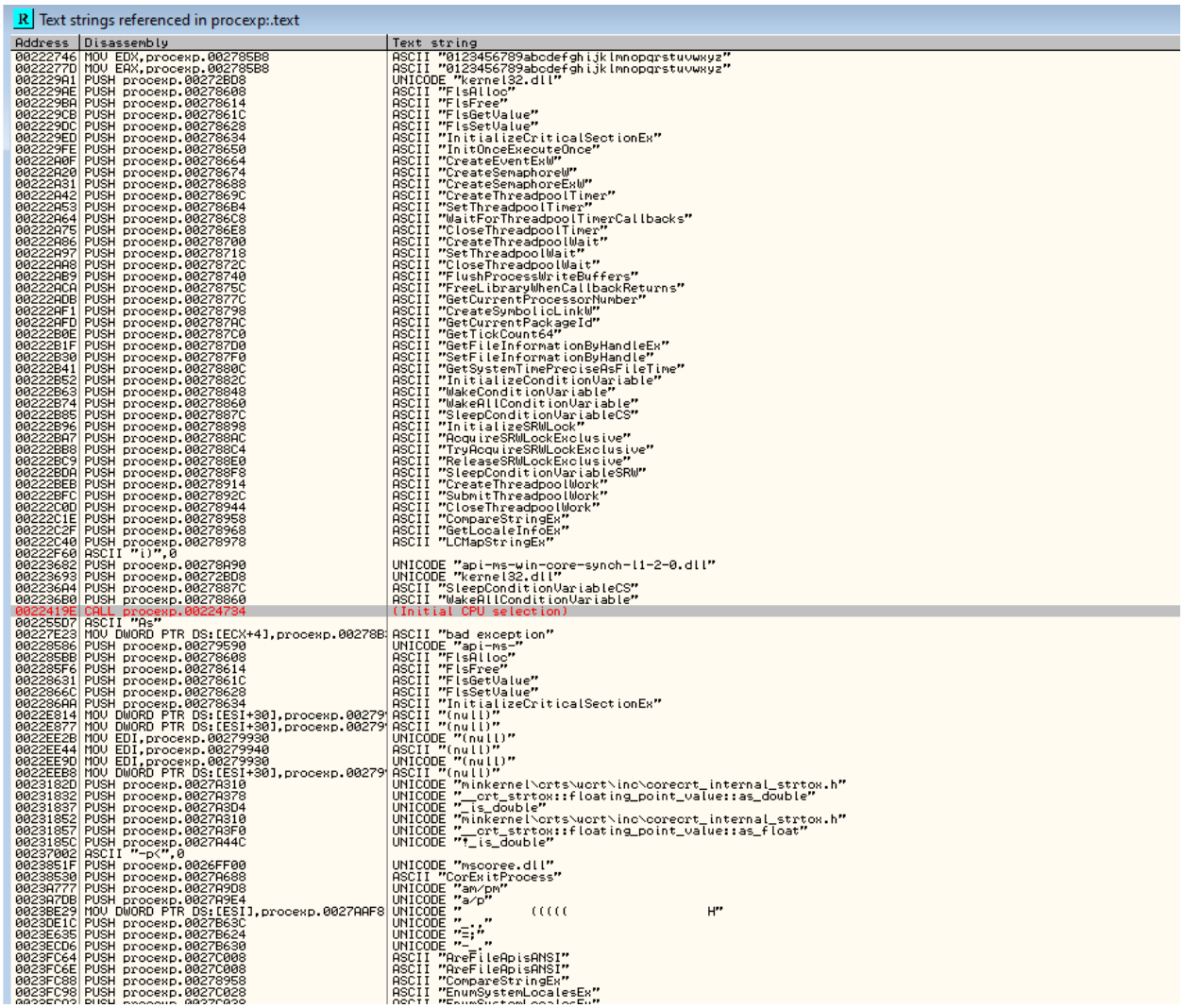
Nous avons ouvert l'exécutable avec Ollydbg afin d'observer les appels système. Nous avons pu voir que l'exécutable appel des fonctions liées à la manipulation des registres Windows, mais en seulement quelques heures de cours nous n'avons pas pu pousser notre investigation davantage, pour ce faire nous aurions dû essayer de définir des breakpoints afin d'analyser les valeurs des registres lors de l'appel des fonctions liés au registres, prises en screenshots ci-dessous.



Addresse	Type	Ordinal	Symbole
0025B1E4	Importe		RectInRegion
0025B708	Importe		RegisterWindowMessageW
0025B778	Importe		AllowSetForegroundWindow
0025B880	Importe		UnregisterClassW
0025B8FC	Importe		SetForegroundWindow
0025B94C	Importe		RegisterClassExW
0025B06C	Importe		RegConnectRegistryW
0025B084	Importe		RegEnumKeyExW
0025B0BC	Importe		RegGetValueW
0025B0C0	Importe		RegCreateKeyW
0025B0CC	Importe		RegCreateKeyExW
0025B0D0	Importe		RegDeleteKeyW
0025B0D4	Importe		RegEnumKeyW
0025B0D8	Importe		RegEnumValueW
0025B0FC	Importe		RegCloseKey
0025B100	Importe		RegDeleteValueW
0025B108	Importe		RegLoadKeyW
0025B10C	Importe		RegOpenKeyW
0025B110	Importe		RegOpenKeyExW
0025B128	Importe		RegQueryValueW
0025B12C	Importe		RegUnLoadKeyW
0025B130	Importe		RegSetValueExW
0025B134	Importe		RegQueryValueExW
0025B138	Importe		RegQueryInfoKeyW

4. Chaînes de caractères utilisés par procexp.exe

À l'aide de Ollydbg, nous avons pu observer les chaînes de caractères utilisées par l'exécutable. Nous n'avons pas pu investiguer davantage leurs buts par manque de temps.



Text strings referenced in procexp:.text

Address	Disassembly	Text string
00222746	MOV EDX,procexp.002785B8	ASCII "0123456789abcdefghijklmnopqrstuvwxyz"
0022277D	MOV EAX,procexp.002785B8	ASCII "0123456789abcdefghijklmnopqrstuvwxyz"
00222901	PUSH procexp.00272BD8	UNICODE "kernel32.dll"
002229AE	PUSH procexp.00278608	ASCII "FlsAlloc"
002229BA	PUSH procexp.00278614	ASCII "FlsFree"
002229CB	PUSH procexp.0027861C	ASCII "FlsGetValue"
002229DC	PUSH procexp.00278628	ASCII "FlsSetValue"
002229ED	PUSH procexp.00278634	ASCII "InitializeCriticalSectionEx"
002229FE	PUSH procexp.00278650	ASCII "InitOnceExecuteOnce"
00222A0F	PUSH procexp.00278664	ASCII "CreateEventExW"
00222A20	PUSH procexp.00278674	ASCII "CreateSemaphoreW"
00222A31	PUSH procexp.00278688	ASCII "CreateSemaphoreExW"
00222A42	PUSH procexp.0027869C	ASCII "CreateThreadpoolTimer"
00222A53	PUSH procexp.002786B4	ASCII "SetThreadpoolTimer"
00222A64	PUSH procexp.002786C8	ASCII "WaitForThreadpoolTimerCallbacks"
00222A75	PUSH procexp.002786E8	ASCII "CloseThreadpoolTimer"
00222A86	PUSH procexp.00278700	ASCII "CreateThreadpoolWait"
00222A97	PUSH procexp.00278718	ASCII "SetThreadpoolWait"
00222AA8	PUSH procexp.0027872C	ASCII "CloseThreadpoolWait"
00222AB9	PUSH procexp.00278740	ASCII "FlushProcessWriteBuffers"
00222ACA	PUSH procexp.0027875C	ASCII "FreeLibraryWhenCallbackReturns"
00222ADB	PUSH procexp.0027877C	ASCII "GetCurrentProcessorNumber"
00222AF1	PUSH procexp.00278798	ASCII "CreateSymbolicLinkW"
00222AFD	PUSH procexp.002787AC	ASCII "GetCurrentPackageId"
00222B0E	PUSH procexp.002787C0	ASCII "GetTickCount64"
00222B1F	PUSH procexp.002787D0	ASCII "GetFileInformationByHandleEx"
00222B30	PUSH procexp.002787F0	ASCII "GetFileInformationByHandle"
00222B41	PUSH procexp.0027880C	ASCII "GetSystemTimePreciseAsFileTime"
00222B52	PUSH procexp.0027882C	ASCII "InitializeConditionVariable"
00222B63	PUSH procexp.00278848	ASCII "WakeConditionVariable"
00222B74	PUSH procexp.00278860	ASCII "WakeAllConditionVariable"
00222B85	PUSH procexp.0027887C	ASCII "SleepConditionVariableCS"
00222B96	PUSH procexp.00278898	ASCII "InitializeSRWLock"
00222BA7	PUSH procexp.002788AC	ASCII "AcquireSRWLockExclusive"
00222BB8	PUSH procexp.002788C4	ASCII "TryAcquireSRWLockExclusive"
00222BC9	PUSH procexp.002788E0	ASCII "ReleaseSRWLockExclusive"
00222BDA	PUSH procexp.002788F8	ASCII "SleepConditionVariableSRW"
00222BEB	PUSH procexp.00278914	ASCII "CreateThreadpoolWork"
00222BFC	PUSH procexp.0027892C	ASCII "SubmitThreadpoolWork"
00222C0D	PUSH procexp.00278944	ASCII "CloseThreadpoolWork"
00222C1E	PUSH procexp.00278958	ASCII "CompareStringEx"
00222C2F	PUSH procexp.00278968	ASCII "GetLocaleInfoEx"
00222C40	PUSH procexp.00278978	ASCII "LCMapStringEx"
00222F60	ASCII "i)",0	
00223682	PUSH procexp.00278A90	UNICODE "api-ms-win-core-synch-l1-2-0.dll"
00223693	PUSH procexp.00272BD8	UNICODE "kernel32.dll"
002236A4	PUSH procexp.0027887C	ASCII "SleepConditionVariableCS"
002236B0	PUSH procexp.00278860	ASCII "WakeAllConditionVariable"
0022419E	CALL procexp.00224734	(Initial CPU selection)
002255D7	ASCII "As"	
00227E23	MOV DWORD PTR DS:[ECX+4],procexp.00278B	ASCII "bad exception"
00228586	PUSH procexp.00279590	UNICODE "api-ms-"
002285BB	PUSH procexp.00278608	ASCII "FlsAlloc"
002285F6	PUSH procexp.00278614	ASCII "FlsFree"
00228631	PUSH procexp.0027861C	ASCII "FlsGetValue"
0022866C	PUSH procexp.00278628	ASCII "FlsSetValue"
002286AA	PUSH procexp.00278634	ASCII "InitializeCriticalSectionEx"
0022E814	MOV DWORD PTR DS:[ESI+30],procexp.00279	ASCII "(null)"
0022E877	MOV DWORD PTR DS:[ESI+30],procexp.00279	ASCII "(null)"
0022EE2B	MOV EDI,procexp.00279930	UNICODE "(null)"
0022EE44	MOV EDI,procexp.00279940	ASCII "(null)"
0022EE9D	MOV EDI,procexp.00279930	UNICODE "(null)"
0022EEB8	MOV DWORD PTR DS:[ESI+30],procexp.00279	ASCII "(null)"
0023182D	PUSH procexp.0027A310	UNICODE "minkernel\crts\ucrt\inc\corecrt_internal_strtox.h"
00231832	PUSH procexp.0027A378	UNICODE "__crt_strtox::floating_point_value::as_double"
00231837	PUSH procexp.0027A3D4	UNICODE "_is_double"
00231852	PUSH procexp.0027A310	UNICODE "minkernel\crts\ucrt\inc\corecrt_internal_strtox.h"
00231857	PUSH procexp.0027A3F0	UNICODE "__crt_strtox::floating_point_value::as_float"
0023185C	PUSH procexp.0027A44C	UNICODE "f_is_double"
00237002	ASCII "-p(",0	
0023851F	PUSH procexp.0026FF00	UNICODE "mscoree.dll"
00238530	PUSH procexp.0027A688	ASCII "CorExitProcess"
0023A777	PUSH procexp.0027A9D8	UNICODE "am/pm"
0023A7DB	PUSH procexp.0027A9E4	UNICODE "a/p"
0023BE29	MOV DWORD PTR DS:[ESI],procexp.0027AAF8	UNICODE " (((((H"
0023DE1C	PUSH procexp.0027B63C	UNICODE "_:,"
0023E635	PUSH procexp.0027B624	UNICODE "=;"
0023ECD6	PUSH procexp.0027B630	UNICODE "-_."
0023FC64	PUSH procexp.0027C008	ASCII "AreFileApisANSI"
0023FC6E	PUSH procexp.0027C008	ASCII "AreFileApisANSI"
0023FC88	PUSH procexp.00278958	ASCII "CompareStringEx"
0023FC98	PUSH procexp.0027C028	ASCII "EnumSystemLocalesEx"
0023FCA2	PUSH procexp.0027C038	ASCII "EnumSystemLocalesEx"

B. Analyse de AffichezMoi

1. Analyse de la chaîne des caractères

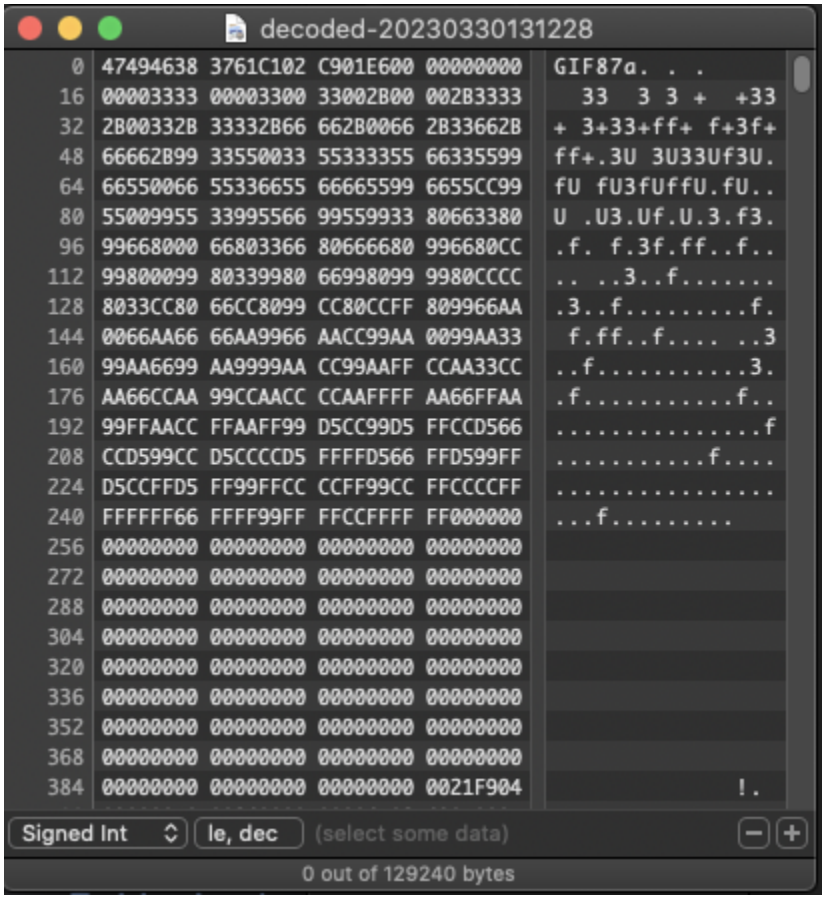
Après un moment de réflexion, un membre du groupe a remarqué que cette chaîne de caractères était encodée en base64 (voir annexe 7.1).

Il en a déduit cela du fait que certains signes ne sont pas présents dans cette chaîne de caractères.

Une absence d’information est elle-même une information.

2. Analyse du fichier décrypté

A l'aide d'un éditeur hexadécimal, ou même d'un éditeur de texte classique, nous pouvons observer l'entête "GIF87" qui nous indique que le fichier est probablement un GIF. Nous avons donc ajouté l'extension .gif au fichier et nous avons pu l'ouvrir avec succès.



3. Résultat trouvé

La chaîne de caractères une fois décodée nous donne cette image GIF de campagne provençale.

6. Conclusion

En conclusion, l'utilisation d'outils comme Sysinternals peut permettre d'accéder à des informations précieuses sur les fichiers, telles que la signature du hash pour vérifier leur intégrité. L'analyse de la chaîne de caractères peut également révéler des informations cachées, il est important de noter que lorsqu'il est difficile de trouver des informations, l’absence de celles-ci peut en soit être une piste.

Dans le cas présent, le décryptage de la chaîne de caractères a permis de découvrir une image de campagne provençale. Ces exemples montrent l'importance de l'analyse approfondie des données ainsi que le fait qu’il est important de garder une image vaste du sujet pour obtenir des informations précises et utiles.

C. Annexes

1. Base64

Index	Binary	Char	Index	Binary	Char	Index	Binary	Char	Index	Binary	Char
0	000000	A	16	010000	Q	32	100000	g	48	110000	w
1	000001	B	17	010001	R	33	100001	h	49	110001	x
2	000010	C	18	010010	S	34	100010	i	50	110010	y
3	000011	D	19	010011	T	35	100011	j	51	110011	z
4	000100	E	20	010100	U	36	100100	k	52	110100	0
5	000101	F	21	010101	V	37	100101	l	53	110101	1
6	000110	G	22	010110	W	38	100110	m	54	110110	2
7	000111	H	23	010111	X	39	100111	n	55	110111	3
8	001000	I	24	011000	Y	40	101000	o	56	111000	4
9	001001	J	25	011001	Z	41	101001	p	57	111001	5
10	001010	K	26	011010	a	42	101010	q	58	111010	6
11	001011	L	27	011011	b	43	101011	r	59	111011	7
12	001100	M	28	011100	c	44	101100	s	60	111100	8
13	001101	N	29	011101	d	45	101101	t	61	111101	9
14	001110	O	30	011110	e	46	101110	u	62	111110	+
15	001111	P	31	011111	f	47	101111	v	63	111111	/

2. Glossaire

Sysinternals : Sysinternals est une suite d'outils système pour Windows créée par Mark Russinovich et Bryce Cogswell. Elle est maintenant développée et maintenue par Microsoft. Les outils Sysinternals sont utilisés pour la gestion et le dépannage des systèmes informatiques sous Windows.

Hash : Un hash, ou fonction de hachage, est un algorithme qui transforme des données (texte, fichier, etc.) en une chaîne de caractères unique et de longueur fixe, appelée empreinte. Les fonctions de hachage sont utilisées pour stocker et comparer des mots de passe, vérifier l'intégrité des données et dans de nombreuses autres applications.

Base64 : Base64 est un système de codage qui permet de représenter des données binaires (octets) sous forme de texte ASCII. Le codage Base64 utilise 64 caractères différents (les lettres majuscules et minuscules, les chiffres et les symboles + et /) pour représenter 6 bits de données binaires.

Hexadécimal : Le système hexadécimal est un système de numération en base 16. Il utilise 16 symboles différents (0-9 et A-F) pour représenter des nombres. Le système hexadécimal est souvent utilisé en informatique pour représenter des adresses mémoire, des couleurs et d'autres valeurs numériques.

ASCII : ASCII est un code de caractères qui associe un nombre unique à chaque caractère utilisé en anglais (lettres majuscules et minuscules, chiffres, ponctuation et caractères spéciaux). Les caractères ASCII sont représentés par des valeurs numériques de 0 à 127.

Unicode : Unicode est un standard de codage de caractères qui permet de représenter des caractères de tous les systèmes d'écriture du monde entier, y compris les alphabets non latins, les symboles et les emojis. Unicode utilise des codes numériques de 0 à plus de 1 million pour représenter chaque caractère.