



Swascan
TINEXTA GROUP

Cactus Ransomware: analisi malware

www.swascan.com

Elementi importanti dell'analisi:

- Attack vector con vulnerabilità Fortinet VPN
- Auto-encryption del ransomware stesso per effettuare bypassing
- Cambio dinamico e consecutivo delle estensioni dei files criptati
- UPX packing
- Algoritmi OpenSSL, AES OCB, ChaCha20_Poly1305
- Scheduled tasks
- Esecuzioni di restart management
- Enumerazioni shares di rete
- Utilizzo del file C:\ProgramData\ntuser.dat per la chiave pubblica dell'auto-encryption
- Crittografia dei files in "buffers"

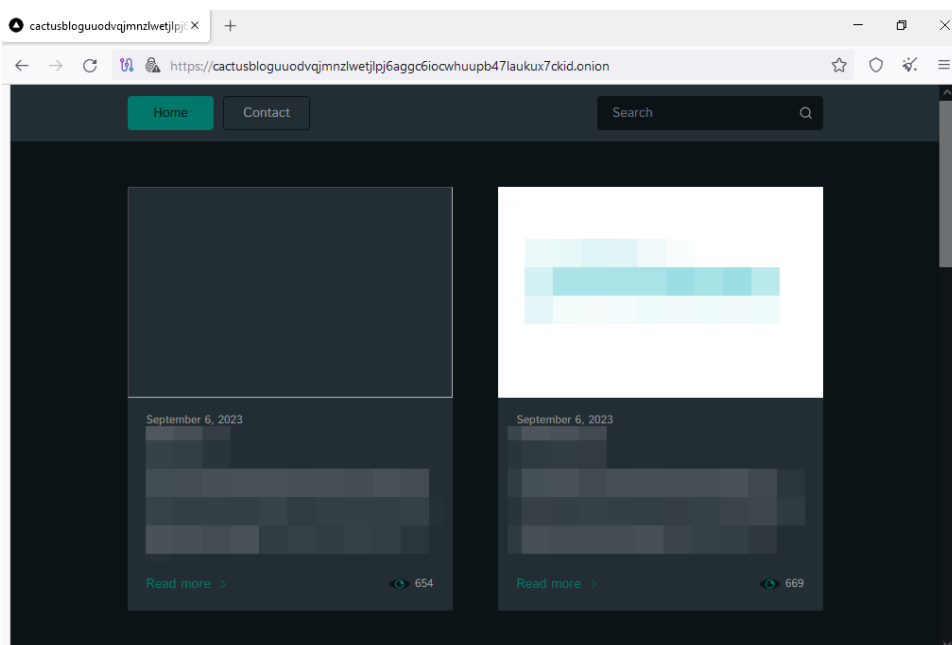
Introduzione.....	3
Analisi statica ed assessment.....	6
Analisi dinamica e disassembling	21
Sessione di debugging.....	39
Threat research	45
IOCs	50
Regola YARA.....	51
Conclusioni	52
Riferimenti	52

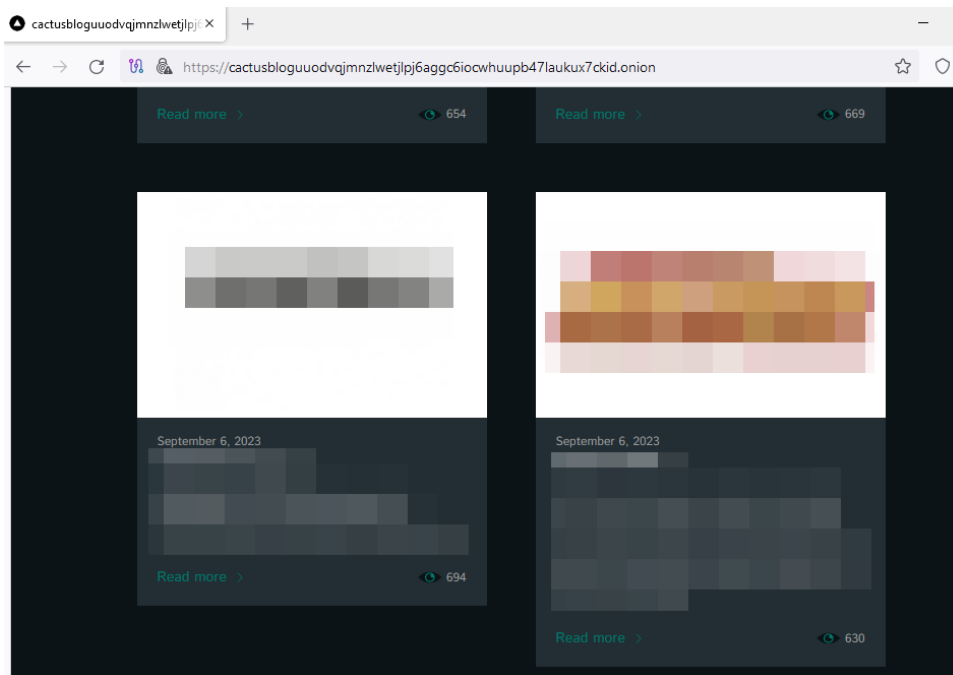
Introduzione

Cactus Ransomware è una nuova minaccia, identificata per la prima volta nel Marzo 2023, con alcune caratteristiche particolari. Viene distribuita nelle infrastrutture compromesse utilizzando principalmente come attack vector alcune vulnerabilità di Fortinet VPN che permettono un accesso non autorizzato. La caratteristica principale di questo ransomware è l'auto-encryption, ovvero la cifratura del medesimo avviene contestualmente alla fase di deployment e questo comporta il fatto che il threat non possa essere rilevato facilmente da EDR, XDR e antimalware. Durante il processo di encryption dei files target vengono modificate in modo dinamico le estensioni utilizzate, passando da `.cts0` a `.cts1`. Durante la fase di esecuzione il malware controlla, similmente a ciò che avviene in un accesso concorrentiale, se un file è accessibile in un determinato momento o meno.

Il sample possiede il portale di data leak

`hxxps[://]cactusbloguodvqjmnzlwetjlpj6aggc6iocwhuupb47laukux7ckid[.]onion`, il quale contiene i dettagli delle varie vittime e dei dati ad esse relativi.





I canali di comunicazione dei malcoders risultano essere principalmente l'indirizzo e-mail `cactus[.]mexicomail[.]com` e la TOX Chat avente URL `hxxps[://]tox[.]chat[/]:/7367B422CD7498D5F2AAF33F58F67A332F8520CF0279A5FBB4611E0121AE421AE1D49ACEABB254686`

Dando uno sguardo approfondito alle modalità di infezione e deployment di Cactus Ransomware appare evidente che esso predilige perlopiù aziende di grandi dimensioni e con un elevato fatturato annuale, con lo scopo, molto probabilmente, di massimizzare il guadagno illecito derivante dalle azioni di riscatto. Quattro grandi aziende recentemente colpite da infezioni Cactus Ransomware fanno riferimento ad attività di private equity, produzione di attrezzi idraulici e cilindri pneumatici, produttori di porte e cancelli, nonché di riscossione debiti. Altre aziende vittime meno recenti riguardano i settori di: agenzie immobiliari, formazioni specialistiche sui trasporti, industry mining e financing, produttori di rimorchi e trasformazione alimentare.

Facendo riferimento a scenari real-world si ha contezza delle seguenti peculiarità chiave:

- Utilizzo dei tools seguenti: Chisel (per effettuare connessioni C&C ed eseguire comandi a bordo delle macchine infette in modalità stealth), Rclone, TotalExec
- Utilizzo del tool SoftPerfect Network Scanner per effettuare discovery dell'infrastruttura target

- Comandi PowerShell per identificare gli hosts di dominio, eventi di login 4624 per identificare le varie utenze
- I tools di remote management maggiormente utilizzati da Cactus Ransomware sono i seguenti: Splashtop, AnyDesk e SuperOps RMM
- Esecuzioni di scripts batch con il fine di disinstallare security solutions ed antimalware products
- Dumping di credenziali LSASS con il fine di effettuare tasks di privilege escalation all'interno del dominio

[1]

Matrice MITRE:

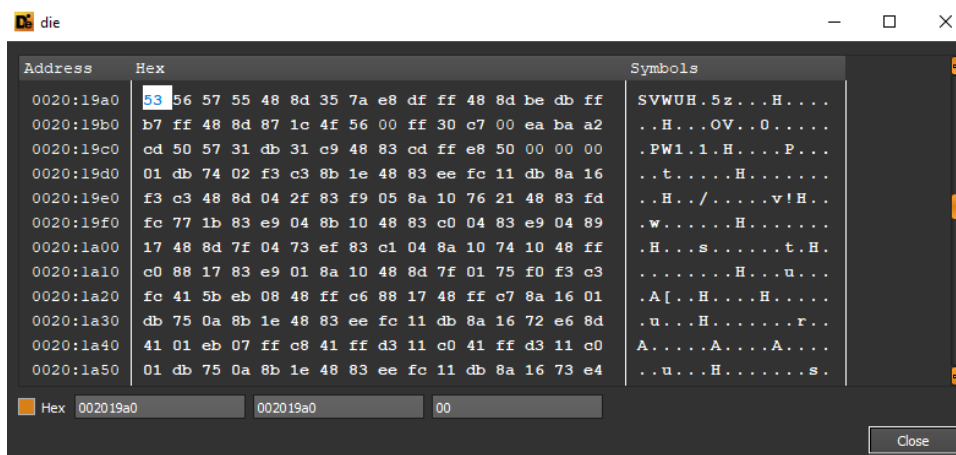
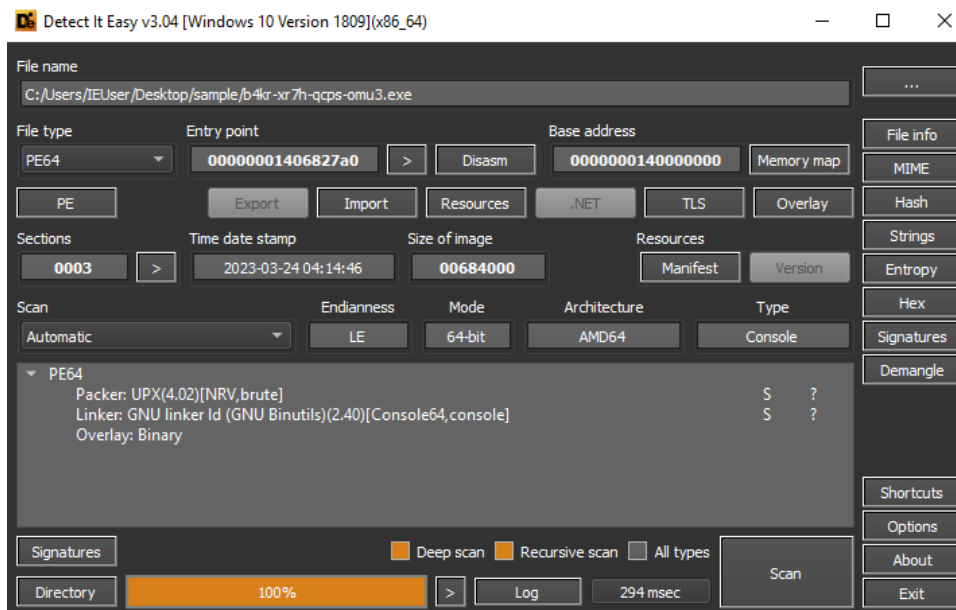
Mitre Att&ck Matrix

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command and Control	Network Effects	Remote Service Effects	Impact
Valid Accounts	Windows Management Instrumentation	Path Interception	1 Process Injection	1 Software Packing	OS Credential Dumping	1 System Information Discovery	Remote Services	Data from Local System	Exfiltration Over Other Network Medium	Data Obfuscation	Eavesdrop on Insecure Network Communication	Remotely Track Device Without Authorization	Modify System Partition
Default Accounts	Scheduled Task/Job	Boot or Logon Initialization Scripts	Boot or Logon Initialization Scripts	1 Process Injection	LSASS Memory	Application Window Discovery	Remote Desktop Protocol	Data from Removable Media	Exfiltration Over Bluetooth	Junk Data	Exploit SS7 to Redirect Phone Calls/SMS	Remotely Wipe Data Without Authorization	Device Lockout
Domain Accounts	At (Linux)	Logon Script (Windows)	Logon Script (Windows)	1 Obfuscated Files or Information	Security Account Manager	Query Registry	SMB/Windows Admin Shares	Data from Network Shared Drive	Automated Exfiltration	Steganography	Exploit SS7 to Track Device Location	Obtain Device Cloud Backups	Delete Device Data

Analisi statica ed assessment

Il sample sottoposto ad analisi ha come hash

78c16de9fc07f1d0375a093903f86583a4e32037a7da8aa2f90ecb15c4862c17, esso è in stato di packed con il packer *UPX*.



Di seguito i dettagli delle sezioni del file, ove si evincono le caratteristiche delle sezioni relative al packer UPX e *.rsrc*.

Memory map

Type: PE64

File offset: 0000000000000000
 Virtual address: 0000000140000000
 Relative virtual address: 0000000000000000

Mode: 64-bit, Endianness: LE, Architecture: AMD64

Offset	Address	Size	Name
0000000000000000	0000000140000000	0000000000000200	PE Header
fffffffffffffff	0000000140000200	0000000000000e00	PE Header
fffffffffffffff	0000000140001000	00000000000480000	Section(0)['UPX0']
0000000000000200	0000000140481000	00000000000201c00	Section(1)['UPX1']
fffffffffffffff	0000000140682c00	0000000000000400	Section(1)['UPX1']
00000000000201e00	0000000140683000	0000000000000800	Section(2)['.rsrc']
fffffffffffffff	0000000140683800	0000000000000800	Section(2)['.rsrc']
00000000000202600	fffffffffffffff	0000000000027c9a7	Overlay

Hex

Address	Hex	Symbols
0000:0000	4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00	MZ.....
0000:0010	b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00@.....
0000:0020	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000:0030	00 00 00 00 00 00 00 00 00 00 00 00 80 00 00 00
0000:0040	0e 1f ba 0e 00 b4 09 cd 21 b8 01 4c cd 21 54 68!.!.!.Th
0000:0050	69 73 20 70 72 6f 67 72 61 6d 20 63 61 6e 6e 6f	is program canno
0000:0060	74 20 62 65 20 72 75 6e 20 69 6e 20 44 4f 53 20	t be run in DOS
0000:0070	6d 6f 64 65 2e 0d 0d 0a 24 00 00 00 00 00 00 00	mode...\$.....
0000:0080	50 45 00 00 64 86 03 00 26 86 1d 64 00 e4 62 00	PE..d...&..d..b.
0000:0090	19 fe 00 00 fd 00 26 00 0b 02 02 28 00 20 20 00&....(.
0000:00a0	00 10 00 00 00 00 48 00 a0 27 68 00 00 10 48 00H..'h...H.

Memory map

Type: PE64

File offset: 0000000000000200
 Virtual address: 0000000140481000
 Relative virtual address: 00000000000481000

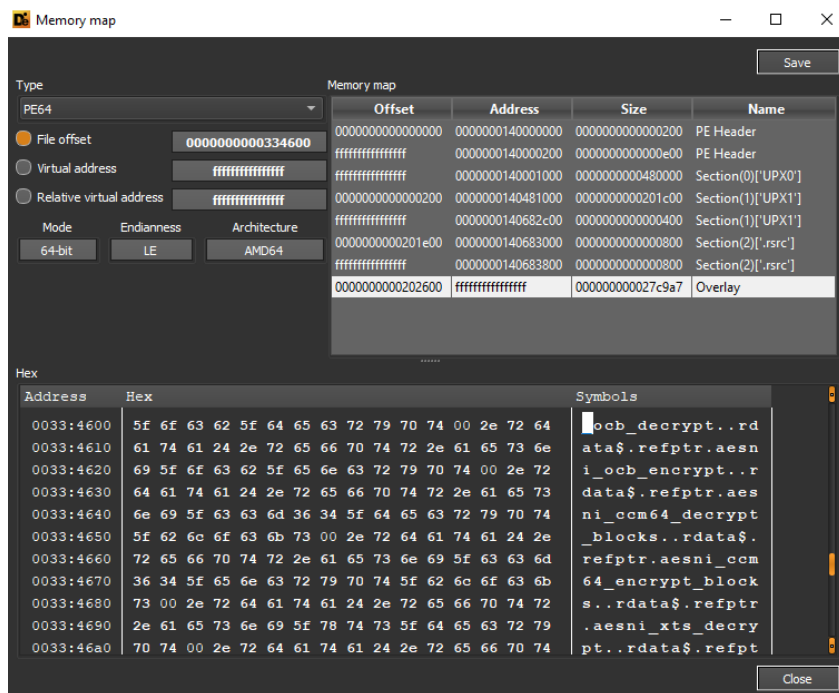
Mode: 64-bit, Endianness: LE, Architecture: AMD64

Offset	Address	Size	Name
0000000000000000	0000000140000000	0000000000000200	PE Header
fffffffffffffff	0000000140000200	0000000000000e00	PE Header
fffffffffffffff	0000000140001000	00000000000480000	Section(0)['UPX0']
0000000000000200	0000000140481000	00000000000201c00	Section(1)['UPX1']
fffffffffffffff	0000000140682c00	0000000000000400	Section(1)['UPX1']
00000000000201e00	0000000140683000	0000000000000800	Section(2)['.rsrc']
fffffffffffffff	0000000140683800	0000000000000800	Section(2)['.rsrc']
00000000000202600	fffffffffffffff	0000000000027c9a7	Overlay

Hex

Address	Hex	Symbols
0000:0200	34 2e 30 32 00 55 50 58 21 0d 24 08 09 1b 4c 7a	4.02.UPX!.\$...Lz
0000:0210	e1 dd ac 8a c8 b1 03 68 00 7a 17 20 00 a7 ad 8ah.z....
0000:0220	00 00 00 00 92 ff bf 29 ff c3 66 66 2e 0f 1f 84)..ff....
0000:0230	00 0e 40 00 48 83 ec 28 48 8b 05 75 91 4b 00 31	..@.H..(H..u.K.1
0000:0240	e4 d8 79 f7 c9 c7 00 01 2c 1c 76 19 79 e6 ff 37	..y.....v.y..7
0000:0250	07 fc 90 66 81 38 4d 5a 75 0f 48 63 50 3c 48 01	...f.8MZu.HcP<H.
0000:0260	d0 ff ff 79 db 50 45 34 74 66 52 1f 89 0d a5 1f	...y.PE4tFR....
0000:0270	52 00 8b 00 85 c0 74 43 b9 02 9f cd 6c 7f 68 e8	R.....tC....l.h.
0000:0280	e1 77 33 08 84 70 42 15 fd 8b 12 ec 1e 64 d8 89	..w3..pB.....d..
0000:0290	10 1f cd 24 60 32 bd cd 84 e9 7e c1 df 83 38 01	...\$`2....~...8.
0000:02a0	74 50 31 c0 32 c4 28 c3 90 b9 01 85 7e ae 6d 6f	tP1.2.(.....~.mo

All'interno della sezione *Overlay*, che contiene dati "appended" all'immagine del Portable Executable, vi sono riferimenti a funzioni di encryption come, ad esempio, *ocb_decrypt* e la sezione *.rdata*.

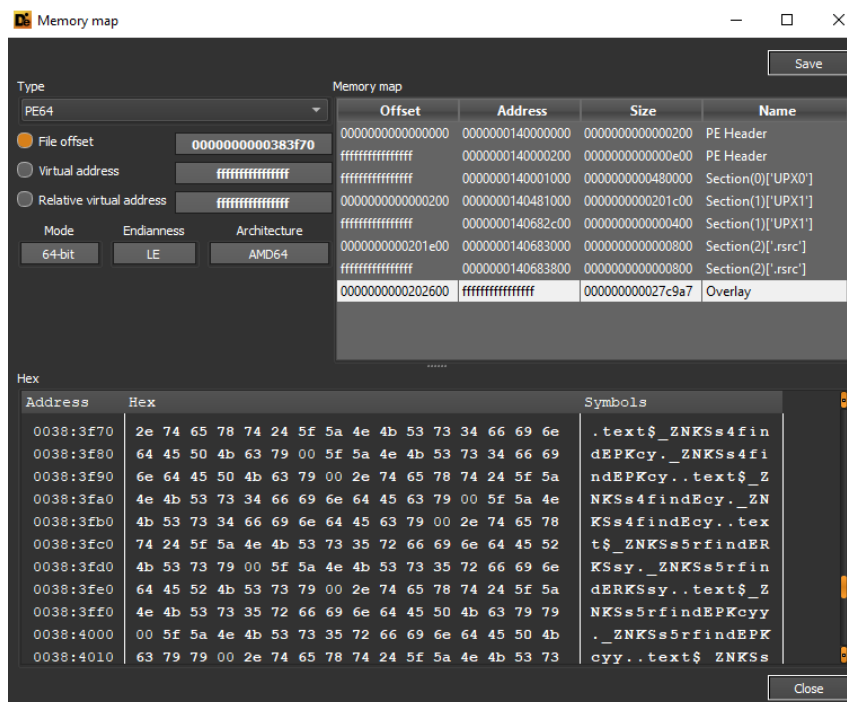


Memory map tool showing PE64 file offset 000000000334600. The memory map table is as follows:

Offset	Address	Size	Name
0000000000000000	0000000140000000	000000000000200	PE Header
fffffffffffffff	0000000140000200	000000000000e00	PE Header
fffffffffffffff	0000000140001000	0000000000480000	Section(0)['.UPX0']
0000000000000200	0000000140481000	0000000000201c00	Section(1)['.UPX1']
fffffffffffffff	0000000140682c00	0000000000000400	Section(1)['.UPX1']
0000000000201e00	0000000140683000	0000000000000800	Section(2)['.rsrc']
fffffffffffffff	0000000140683800	0000000000000800	Section(2)['.rsrc']
0000000000202600	fffffffffffffff	000000000027c9a7	Overlay

Hex dump starting at 0033:4600:

Address	Hex	Symbols
0033:4600	5f 6f 63 62 5f 64 65 63 72 79 70 74 00 2e 72 64	ocb_decrypt..rd
0033:4610	61 74 61 24 2e 72 65 66 70 74 72 2e 61 65 73 6e	ata\$.refptr.aesni
0033:4620	69 5f 6f 63 62 5f 65 6e 63 72 79 70 74 00 2e 72	i_ocb_encrypt..r
0033:4630	64 61 74 61 24 2e 72 65 66 70 74 72 2e 61 65 73	data\$.refptr.aes
0033:4640	6e 69 5f 63 63 6d 36 34 5f 64 65 63 72 79 70 74	ni_ccm64_decrypt
0033:4650	5f 62 6c 6f 63 6b 73 00 2e 72 64 61 74 61 24 2e	_blocks..rdata\$.
0033:4660	72 65 66 70 74 72 2e 61 65 73 6e 69 5f 63 63 6d	refptr.aesni_ccm
0033:4670	36 34 5f 65 6e 63 72 79 70 74 5f 62 6c 6f 63 6b	64_encrypt_block
0033:4680	73 00 2e 72 64 61 74 61 24 2e 72 65 66 70 74 72	s..rdata\$.refptr
0033:4690	2e 61 65 73 6e 69 5f 78 74 73 5f 64 65 63 72 79	.aesni_xts_decryp
0033:46a0	70 74 00 2e 72 64 61 74 61 24 2e 72 65 66 70 74	pt..rdata\$.refptr



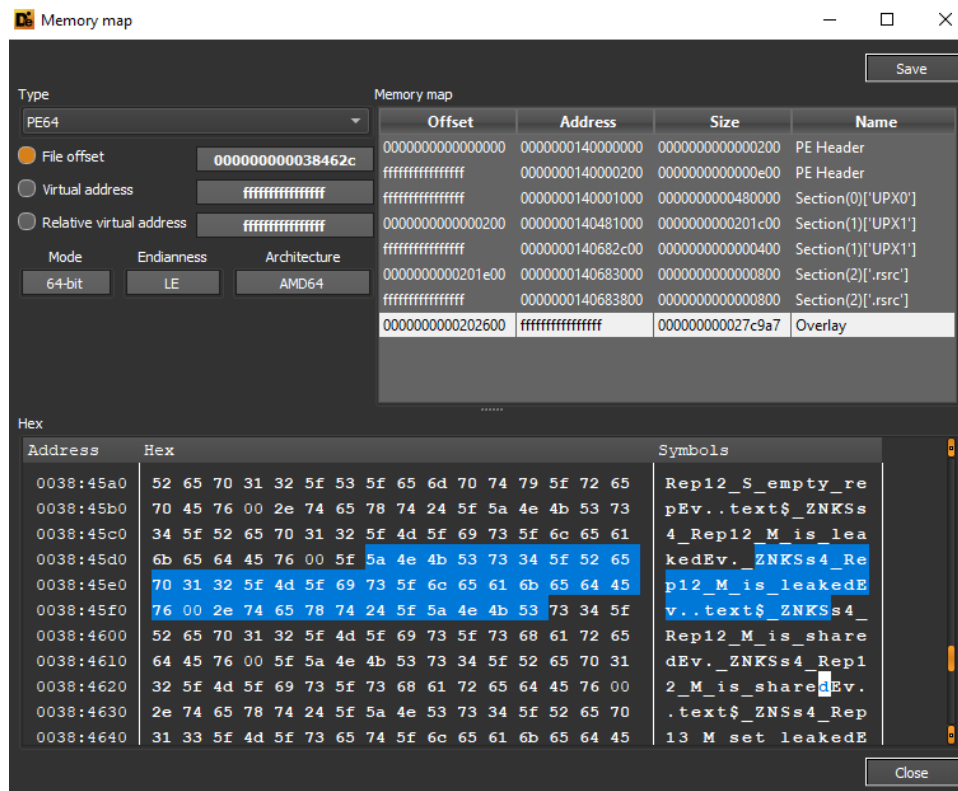
Memory map tool showing PE64 file offset 000000000383f70. The memory map table is as follows:

Offset	Address	Size	Name
0000000000000000	0000000140000000	000000000000200	PE Header
fffffffffffffff	0000000140000200	000000000000e00	PE Header
fffffffffffffff	0000000140001000	0000000000480000	Section(0)['.UPX0']
0000000000000200	0000000140481000	0000000000201c00	Section(1)['.UPX1']
fffffffffffffff	0000000140682c00	0000000000000400	Section(1)['.UPX1']
0000000000201e00	0000000140683000	0000000000000800	Section(2)['.rsrc']
fffffffffffffff	0000000140683800	0000000000000800	Section(2)['.rsrc']
0000000000202600	fffffffffffffff	000000000027c9a7	Overlay

Hex dump starting at 0038:3f70:

Address	Hex	Symbols
0038:3f70	2e 74 65 78 74 24 5f 5a 4e 4b 53 73 34 66 69 6e	.text\$_ZNKSs4fin
0038:3f80	64 45 50 4b 63 79 00 5f 5a 4e 4b 53 73 34 66 69	dEPKcy._ZNKSs4fi
0038:3f90	6e 64 45 50 4b 63 79 00 2e 74 65 78 74 24 5f 5a	ndEPKcy..text\$_Z
0038:3fa0	4e 4b 53 73 34 66 69 6e 64 45 63 79 00 5f 5a 4e	NKSs4findEcy._ZN
0038:3fb0	4b 53 73 34 66 69 6e 64 45 63 79 00 2e 74 65 78	KSs4findEcy..tex
0038:3fc0	74 24 5f 5a 4e 4b 53 73 35 72 66 69 6e 64 45 52	t\$_ZNKSs5rfindER
0038:3fd0	4b 53 73 79 00 5f 5a 4e 4b 53 73 35 72 66 69 6e	KSSy._ZNKSs5rfin
0038:3fe0	64 45 52 4b 53 73 79 00 2e 74 65 78 74 24 5f 5a	dBRKSsy..text\$_Z
0038:3ff0	4e 4b 53 73 35 72 66 69 6e 64 45 50 4b 63 79 79	NKSs5rfindEPKcy
0038:4000	00 5f 5a 4e 4b 53 73 35 72 66 69 6e 64 45 50 4b	._ZNKSs5rfindEPK
0038:4010	63 79 79 00 2e 74 65 78 74 24 5f 5a 4e 4b 53 73	cyy..text\$_ZNKSs

Sempre all'interno dell'Overlay data vi sono dettagli inerenti a quello che sembrerebbe essere un attributo booleano di controllo in merito ad un contesto di leaking.



Relativamente agli imports, possiamo evidenziare la libreria *ADVAPI32.dll* (per eseguire la funzione di event log appending *ReportEventW*), *Rstrtmgr.dll* (per eseguire la funzione *RmGetList* per ottenere una lista delle applicazioni e dei servizi che utilizzano le risorse registrate all'interno della sessione di Restart Manager), *WS2_32.dll* e *WSOCK32.dll* (al fine di gestire connessioni tramite sockets).

PE

Hash 64: 000000572c4142e Hash 32: 1fea2626

	ginalFirstTh	neDateStan	rwarderCha	Name	FirstThunk	Hash	
0	00000000	00000000	00000000	00683638	006835a0	7db51e64	ADVAPI32.dll
1	00000000	00000000	00000000	00683645	006835b0	82a048fc	KERNEL32.DLL
2	00000000	00000000	00000000	00683652	006835d8	e8f5db23	msvcrt.dll
3	00000000	00000000	00000000	0068365d	006835e8	d909191e	RstrtlMgr.DLL
4	00000000	00000000	00000000	0068366a	006835f8	a7cddb9	SHELL32.dll
5	00000000	00000000	00000000	00683676	00683608	50c13dd8	USER32.dll
6	00000000	00000000	00000000	00683681	00683618	cae62a6a	WS2_32.dll
7	00000000	00000000	0068368c	00683628	5531f395	WSOCK32.dll	

	Thunk	Ordinal	Hint	Name
0	0000000000683698		0000	ReportEventW

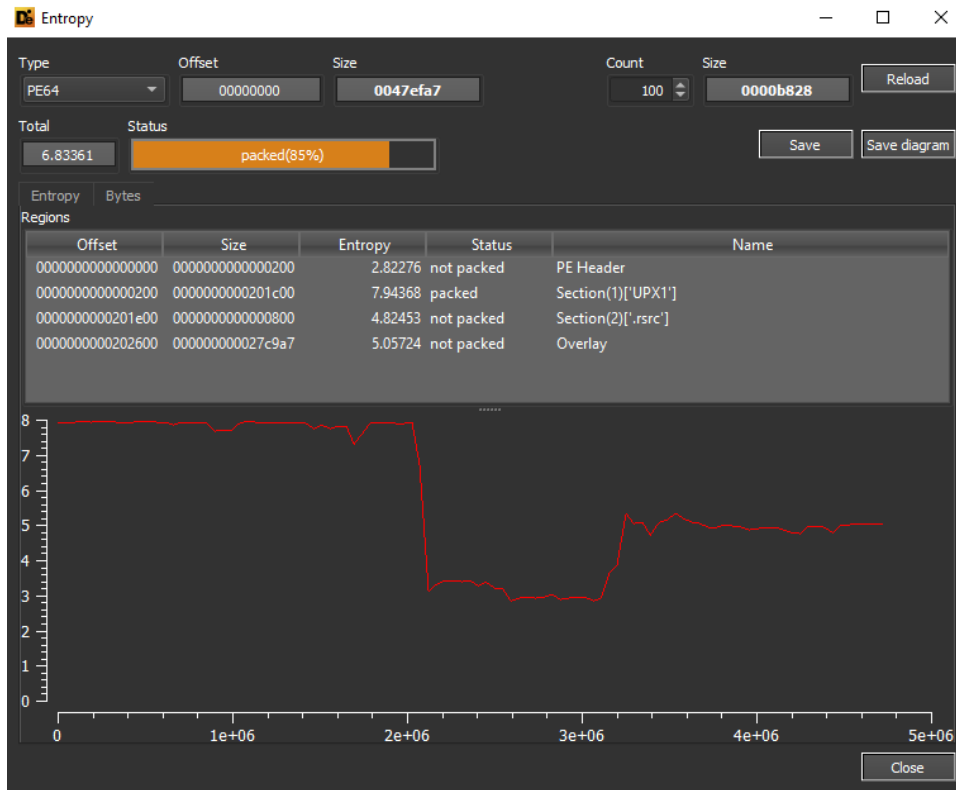
PE

Hash 64: 000000572c4142e Hash 32: 1fea2626

	ginalFirstTh	neDateStan	rwarderCha	Name	FirstThunk	Hash	
0	00000000	00000000	00000000	00683638	006835a0	7db51e64	ADVAPI32.dll
1	00000000	00000000	00000000	00683645	006835b0	82a048fc	KERNEL32.DLL
2	00000000	00000000	00000000	00683652	006835d8	e8f5db23	msvcrt.dll
3	00000000	00000000	00000000	0068365d	006835e8	d909191e	RstrtlMgr.DLL
4	00000000	00000000	00000000	0068366a	006835f8	a7cddb9	SHELL32.dll
5	00000000	00000000	00000000	00683676	00683608	50c13dd8	USER32.dll
6	00000000	00000000	00000000	00683681	00683618	cae62a6a	WS2_32.dll
7	00000000	00000000	00000000	0068368c	00683628	5531f395	WSOCK32.dll

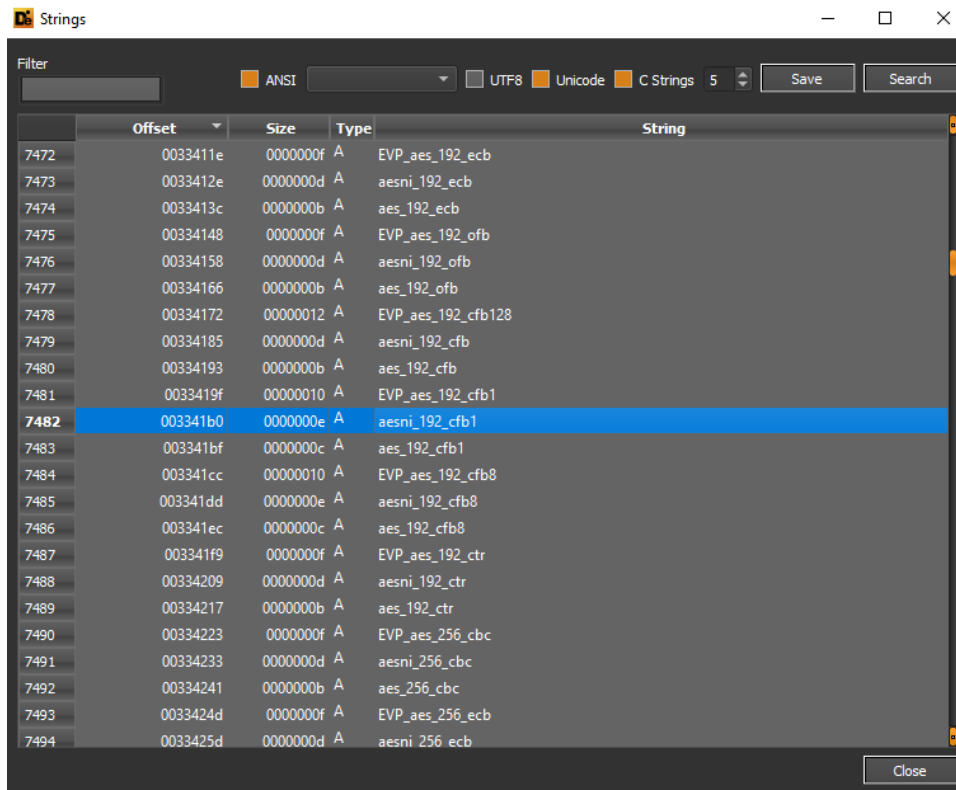
	Thunk	Ordinal	Hint	Name
0	0000000000683712		0000	bind

Il coefficiente d'entropia più alto, dovuto alla fase di UPX packing, si attesta a **7.94368**.

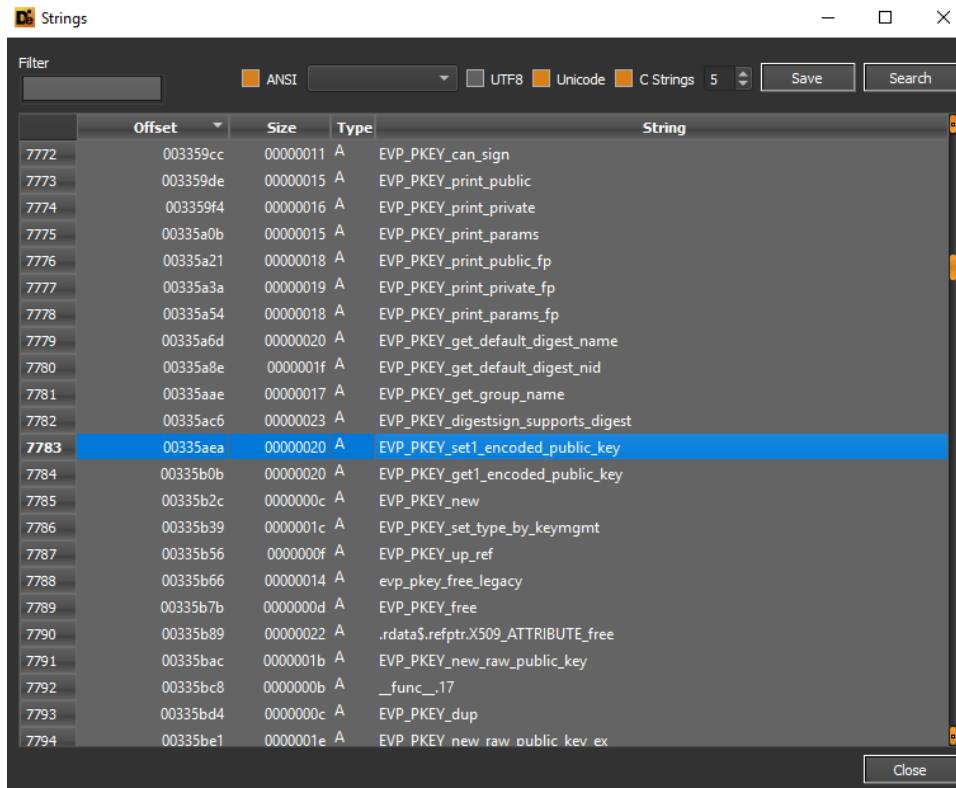


```
File name: C:/Users/IEUser/Desktop/sample/b4kr-xr7h-qcps-omu3.exe
Size: 4714407 (4.50 MB)
MD5: e28db6a65da2ebcf304873c9a5ed086d
SHA1: cb570234349507a204c558fc8c4ecf713e2c0ac3
Entropy: 6.83361 (packed)
Operation system: Windows (Server 2003)
Architecture: AMD64
Mode: 64-bit
Type: Console
Endianness: LE
Entry point (Address): 00000001406827a0
Entry point (Offset): 002019a0
Entry point (Relative address): 006827a0
Entry point (Bytes): 53565755488d357ae8dff488dbedbf7ff488d871c4f5600ff30c700eabaa2cd50
Entry point (Signature): 53565755488d35.....488dbe.....488d87.....ff30c700.....50
Entry point (Signature) (Rel): 53565755488d35.....488dbe.....488d87.....ff30c700.....50
```

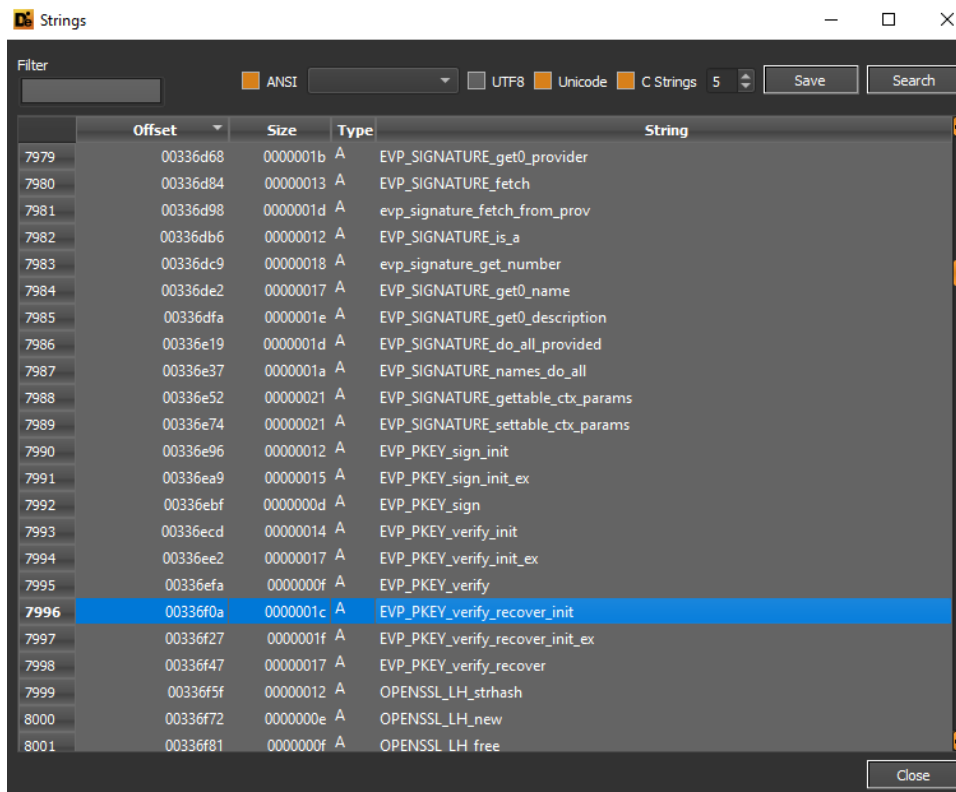
Dalle stringhe estraibili dall'eseguibile si notano riferimenti a *CBC* encryption routines e *CBC* decryption routines relativi all'algoritmo *AES*.



Di seguito i dettagli di alcuni oggetti di public key storage, per esempio *EVP_PKEY_set1_encoded_public_key*:



La funzione *EVP_PKEY_verify_recover_init* può essere utilizzata al fine di verificare la signature associata all'algoritmo per la chiave pubblica.



A seguire un dettaglio relativo all' algoritmo *chacha20_poly1305*, viene evidenziato qui sotto l'oggetto cipher. L'algoritmo in questione è un mix tra *ChaCha20* e *Poly1305* e prevede un processo preliminare di authentication.

Strings

Filter: ANSI UTF8 Unicode C Strings 5 Save Search

	Offset	Size	Type	String
9553	0033f66f	0000000f	A	chacha20_newctx
9554	0033f67f	00000015	A	ossl_chacha20_initctx
9555	0033f695	00000010	A	chacha20_initkey
9556	0033f6a6	0000000f	A	chacha20_initiv
9557	0033f6b6	0000000f	A	chacha20_cipher
9558	0033f6c6	0000001c	A	ossl_prov_cipher_hw_chacha20
9559	0033f6e3	0000000b	A	chacha20_hw
9560	0033f6ef	00000025	A	chacha20_poly1305_gettable_cbx_params
9561	0033f715	0000002b	A	chacha20_poly1305_known_gettable_cbx_params
9562	0033f741	00000020	A	chacha20_poly1305_set_cbx_params
9563	0033f762	00000020	A	chacha20_poly1305_get_cbx_params
9564	0033f783	0000001c	A	chacha20_poly1305_get_params
9565	0033f7a0	00000017	A	chacha20_poly1305_final
9566	0033f7b8	00000018	A	chacha20_poly1305_cipher
9567	0033f7d1	00000017	A	chacha20_poly1305_dinit
9568	0033f7e9	00000017	A	chacha20_poly1305_einit
9569	0033f801	00000018	A	chacha20_poly1305_newctx
9570	0033f81a	00000019	A	chacha20_poly1305_freectx
9571	0033f834	00000018	A	chacha_poly1305_tls_init
9572	0033f84d	00000018	A	chacha20_poly1305_initiv
9573	0033f866	0000001d	A	chacha20_poly1305_aead_cipher
9574	0033f884	00000020	A	chacha_poly1305_tls_iv_set_fixed
9575	0033f8a5	00000019	A	chacha20 poly1305 initkev

Close

```

.text:000000014010DA40 ; DATA XREF: ChaCha20_16x+1424r
.text:000000014010DA40 db 3 dup(0), 5, 3 dup(0), 6, 3 dup(0), 7, 3 dup(0), 8
.text:000000014010DA40 db 3 dup(0), 9, 3 dup(0), 0Ah, 3 dup(0), 0Bh, 3 dup(0)
.text:000000014010DA40 db 0Ch, 3 dup(0), 0Dh, 3 dup(0), 0Eh, 3 dup(0), 0Fh, 3 dup(0)
.text:000000014010DA80 byte_14010DA80 db 10h, 3 dup(0), 10h, 3 dup(0), 10h, 3 dup(0), 10h, 3 dup(0)
.text:000000014010DA80 ; DATA XREF: ChaCha20_16x+1984r
.text:000000014010DA80 db 10h, 3 dup(0), 10h, 3 dup(0), 10h, 3 dup(0), 10h, 3 dup(0)
.text:000000014010DA80 db 10h, 3 dup(0), 10h, 3 dup(0), 10h, 3 dup(0), 10h, 3 dup(0)
.text:000000014010DA80 db 10h, 3 dup(0), 10h, 3 dup(0), 10h, 3 dup(0), 10h, 3 dup(0)
.text:000000014010DAC0 xmmword_14010DAC0 xmmword 6B20657479622D32332064E61707865h
.text:000000014010DAC0 ; DATA XREF: ChaCha20_sse3+524r
.text:000000014010DAC0 ; ChaCha20_128+464r ...
.text:000000014010DAD0 db 0
.text:000000014010DAD1 aChaCha20ForX86 db 'ChaCha20 for x86_64, CRYPTOAGMS by <appro@openssl.org>',0
.text:000000014010DB08 align 40h
.text:000000014010DB40 ; ===== S U B R O U T I N E =====
.text:000000014010DB40
.text:000000014010DB40
.text:000000014010DB40 public ChaCha20_ctr32
.text:000000014010DB40 ChaCha20_ctr32 proc near ; CODE XREF: chacha20_cipher+B61p
.text:000000014010DB40 ; chacha20_cipher+1177p ...
.text:000000014010DB40 var_88 = dword ptr -88h
.text:000000014010DB40 var_84 = dword ptr -84h
.text:000000014010DB40 var_80 = qword ptr -80h
.text:000000014010DB40 var_78 = xmmword ptr -78h
0010D0D1|000000014010DAD1: .text:aChaCha20ForX86|(Synchronized with Hex View-1)

```

```

ChaCha20_128 proc near
var_A8= xmmword ptr -0A8h
var_98= xmmword ptr -98h
var_88= xmmword ptr -88h
var_78= xmmword ptr -78h
arg_0= qword ptr 8
arg_8= qword ptr 10h
arg_20= qword ptr 28h

mov     [rsp+arg_0], rdi
mov     [rsp+arg_8], rsi
mov     rax, rsp
mov     rdi, rcx
mov     rsi, rdx
mov     rdx, r8
mov     rcx, r9
mov     r8, [rsp+arg_20]

loc_14010E1BE:
mov     r9, rsp

```

Nello screenshot sottostante sono invece evidenziate funzioni di loading della chiave privata associata a *OpenSSL*:

	Offset	Size	Type	String
13120	00353357	0000001d	A	ossl_ecx_key_allocate_privkey
13377	00354b1a	00000014	A	openssl_load_privkey
13392	00354c0b	00000020	A	ENGINE_set_load_privkey_function
13395	00354c75	00000020	A	ENGINE_get_load_privkey_function

Qui di seguito la fase di unpacking UPX:

```

C:\Windows\system32>C:\Users\IEUser\Desktop\upx-4.0.2-win64\upx.exe -d C:\Users\IEUser\Desktop\sample\b4kr-xr7h-qcps-omu3.exe

      Ultimate Packer for eXecutables
      Copyright (C) 1996 - 2023
UPX 4.0.2      Markus Oberhumer, Laszlo Molnar & John Reiser   Jan 30th 2023

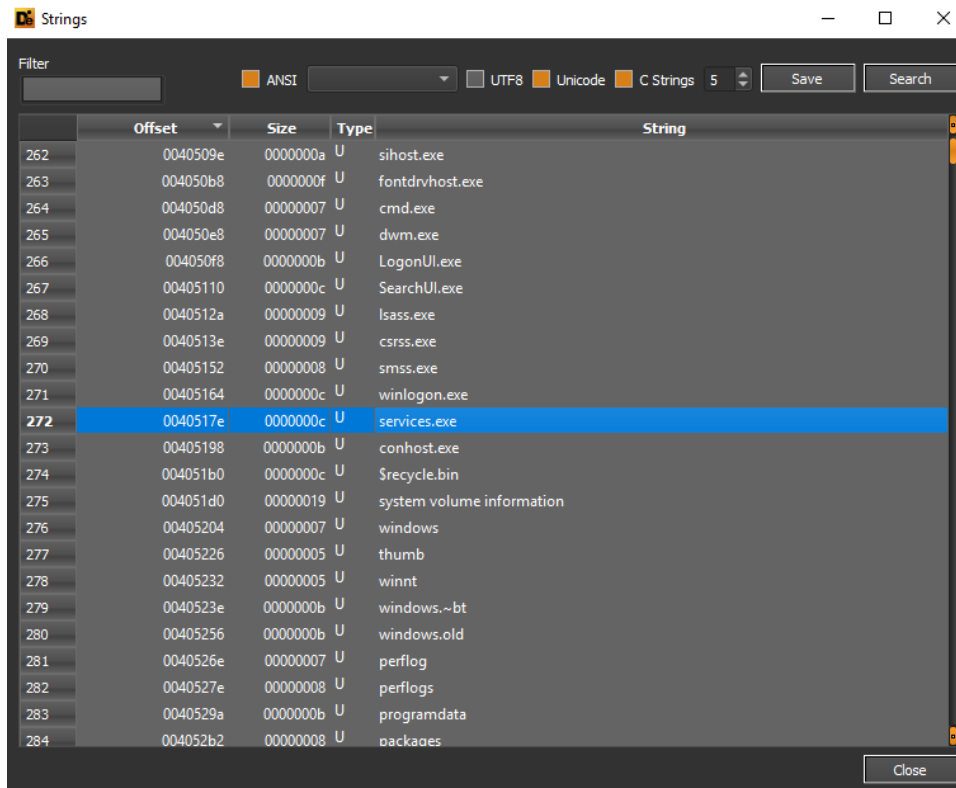
-----
File size      Ratio      Format      Name
-----
9088423 <- 4714407 51.87%    win64/pe   b4kr-xr7h-qcps-omu3.exe

Unpacked 1 file.

C:\Windows\system32>

```

Dalle stringhe estratte vi sono dettagli inerenti a processi di authentication (*lsass.exe* e *winlogon.exe*), *services.exe*, la cartella *ProgramData*.



Cactus Ransomware effettua il bypassing della protection di Windows Defender, utilizza TOR Browser per il contatto con i malcoders, il file *ntuser.dat* (salvato nella cartella C:\ProgramData) viene utilizzato con lo scopo di archiviare la chiave per la crittografia del file eseguibile. Esso non rappresenta il file *ntuser.dat* originale di Windows ma piuttosto un file denominato in tal modo con tutta probabilità per effettuare evasion e confondere la natura del file stesso. Inoltre, lo script batch *rn.bat* viene con tutta probabilità utilizzato per la fase di rinomina di files. Il file di log C:\ProgramData\update.log viene utilizzato per tracciamento dell'infezione ransomware, mentre il threat minaccia anche la pubblicazione dei dati della vittima in caso di mancato pagamento del riscatto.

Strings

Filter

ANSI UTF8 Unicode C Strings 5 Save Search

Offset	Size	Type	String
283	0040529a	0000000b U	programdata
284	004052b2	00000008 U	packages
285	004052cc	0000000b U	windowsapps
286	004052e4	00000009 U	microsoft
287	004052f8	00000010 U	windows defender
288	00405320	00000010 U	microsoft shared
289	00405348	00000011 U	internet explorer
290	0040536c	0000000b U	tor browser
291	00405384	00000006 U	ctsick
292	00405398	00000011 U	CaCtUs.ReAdMe.bt
293	004053bc	0000000b U	desktop.ini
294	004053d4	0000000a U	update.log
295	004053ea	0000000a U	ntuser.dat
296	00405414	00000008 U	\\?\UNC\
297	00405428	00000007 U	\rn.bat
298	00405438	00000031 A	basic_string: construction from null is not valid
299	00405480	0000000b U	%d.%m.%Y %X
300	00405498	00000019 U	C:\ProgramData\update.log
301	004054d0	00000033 U	Your systems were accessed and encrypted by Cactus.
302	00405540	0000005d U	To recover your files and prevent data disclosure contact us via email: ...
303	00405600	0000001a U	Your unique ID reference:
304	00405638	00000028 U	Backup contact: TOX (https://tox.chat/):
305	00405690	0000004c U	7367B422CD7498D5F2AAF33F58F67A332F8520CF0279A5FBB4611E0121AE421AE1D4...

Close

Strings

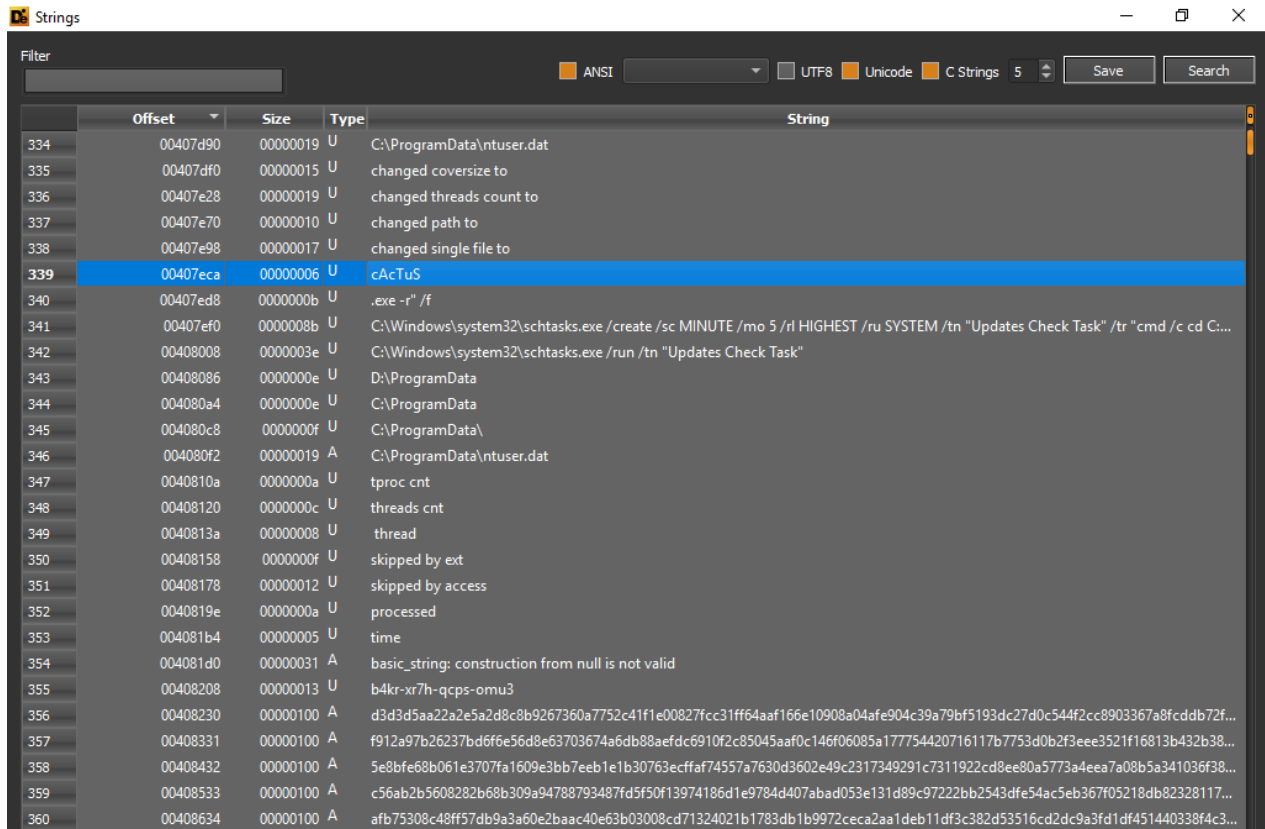
Filter

ANSI UTF8 Unicode C Strings 5 Save Search

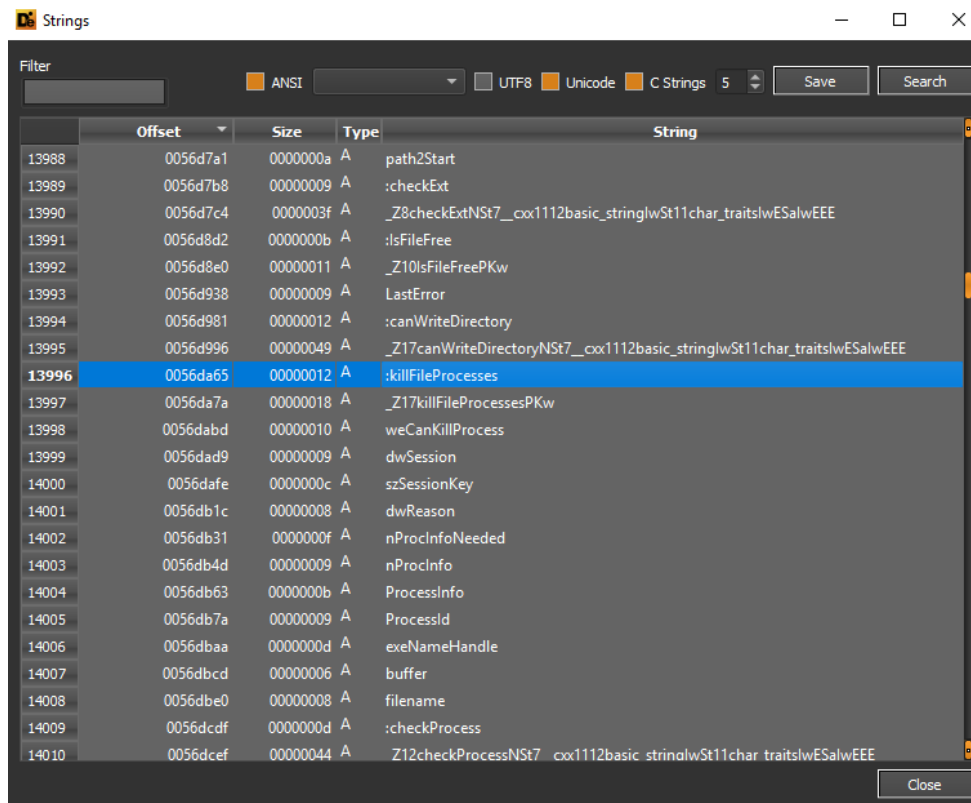
Offset	Size	Type	String
298	00405438	00000031 A	basic_string: construction from null is not valid
299	00405480	0000000b U	%d.%m.%Y %X
300	00405498	00000019 U	C:\ProgramData\update.log
301	004054d0	00000033 U	Your systems were accessed and encrypted by Cactus.
302	00405540	0000005d U	To recover your files and prevent data disclosure contact us via email: ...
303	00405600	0000001a U	Your unique ID reference:
304	00405638	00000028 U	Backup contact: TOX (https://tox.chat/):
305	00405690	0000004c U	7367B422CD7498D5F2AAF33F58F67A332F8520CF0279A5FBB4611E0121AE421AE1D4...
306	00405730	00000100 U	
307	00405932	00000100 U	
308	00405b34	00000100 U	
309	00405d36	00000100 U	
310	00405f38	00000100 U	
311	0040613a	00000100 U	
312	0040633c	00000100 U	
313	0040653e	00000046 U	
314	004065d0	00000100 U	
315	004067d2	00000100 U	
316	004069d4	00000100 U	
317	00406bd6	00000100 U	
318	00406dd8	00000100 U	
319	00406fda	00000100 U	
320	004071dc	00000100 U	E7665737469676174696F6E207768656E20746865207375626A656374206F666207468652...

Close

Cactus Ransomware crea uno scheduled task per la persistenza malevola denominato **“Updates Check Task”**



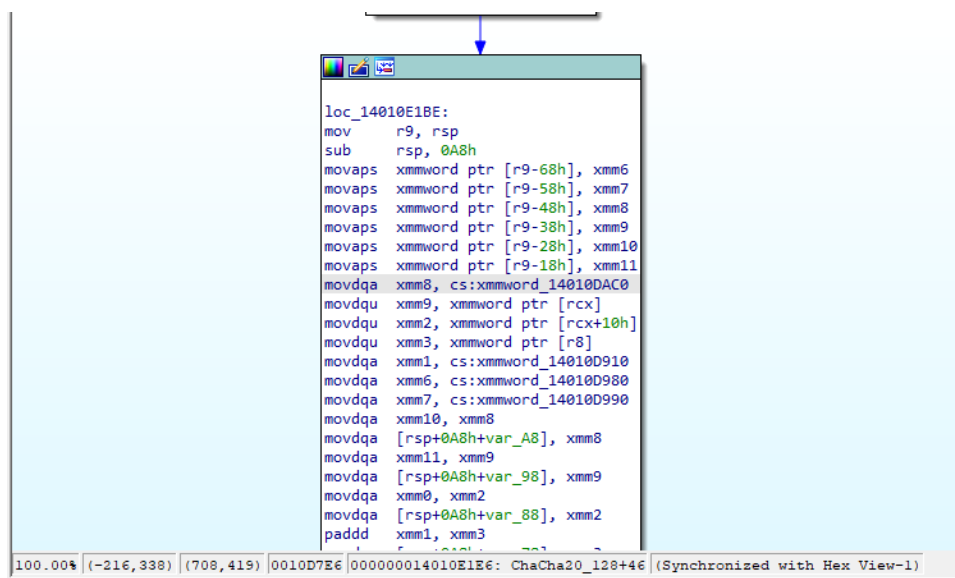
Si notino le funzioni *IsFileFree* (che viene utilizzata al fine di controllare se un determinato file si può criptare), *canWriteDirectory* (per controllare se una determinata cartella permette eventi di scrittura), *killFileProcesses* (con lo scopo di terminare i processi che occupano le risorse di un determinato file), *weCanKillProcess* (utilizzabile al fine di controllare se un determinato processo può essere terminato)



Offset	Size	Type	String
13988	0056d7a1	0000000a A	path2Start
13989	0056d7b8	00000009 A	:checkExt
13990	0056d7c4	0000003f A	_Z8checkExtNSt7__cxx112basic_stringlwSt11char_traitslwESalwEEE
13991	0056d8d2	0000000b A	:IsFileFree
13992	0056d8e0	00000011 A	_Z10IsFileFreePKw
13993	0056d938	00000009 A	LastError
13994	0056d981	00000012 A	:canWriteDirectory
13995	0056d996	00000049 A	_Z17canWriteDirectoryNSt7__cxx112basic_stringlwSt11char_traitslwESalwEEE
13996	0056da65	00000012 A	:killFileProcesses
13997	0056da7a	00000018 A	_Z17killFileProcessesPKw
13998	0056dabd	00000010 A	weCanKillProcess
13999	0056dad9	00000009 A	dwSession
14000	0056dafa	0000000c A	szSessionKey
14001	0056db1c	00000008 A	dwReason
14002	0056db31	0000000f A	nProclInfoNeeded
14003	0056db4d	00000009 A	nProclInfo
14004	0056db63	0000000b A	ProcessInfo
14005	0056db7a	00000009 A	ProcessId
14006	0056dbaa	0000000d A	exeNameHandle
14007	0056dbcd	00000006 A	buffer
14008	0056dbe0	00000008 A	filename
14009	0056dcdf	0000000d A	:checkProcess
14010	0056dcef	00000044 A	Z12checkProcessNSt7__cxx112basic_stringlwSt11char_traitslwESalwEEE

Analisi dinamica e disassembling

Contestualmente ad esecuzioni ChaCha20, all'interno della label `loc_14010E1BE`, vi sono numerose operazioni `movaps`, `movdqa` e `movdqu` inerenti a registri `xmm`



```
loc_14010E1BE:
mov     r9, rsp
sub     rsp, 0A8h
movaps  xmmword ptr [r9-68h], xmm6
movaps  xmmword ptr [r9-58h], xmm7
movaps  xmmword ptr [r9-48h], xmm8
movaps  xmmword ptr [r9-38h], xmm9
movaps  xmmword ptr [r9-28h], xmm10
movaps  xmmword ptr [r9-18h], xmm11
movdqa  xmm8, cs:xmmword_14010DAC0
movdqu  xmm9, xmmword ptr [rcx]
movdqu  xmm2, xmmword ptr [rcx+10h]
movdqu  xmm3, xmmword ptr [r8]
movdqa  xmm1, cs:xmmword_14010D910
movdqa  xmm6, cs:xmmword_14010D980
movdqa  xmm7, cs:xmmword_14010D990
movdqa  xmm10, xmm8
movdqa  [rsp+0A8h+var_A8], xmm8
movdqa  xmm11, xmm9
movdqa  [rsp+0A8h+var_98], xmm9
movdqa  xmm0, xmm2
movdqa  [rsp+0A8h+var_88], xmm2
padd   xmm1, xmm3
```

Cactus Ransomware effettua modifiche in merito al file di configurazione `desktop.ini` conseguentemente alla fase di desktop changing e readme notes dropping, per quanto riguarda il file di tracciamento `update.log` vengono anche aggiunti attributi di newline:

```

var_18= byte ptr -18h
push rbp
push rbx
sub rsp, 58h
lea rbp, [rsp+68h+var_18]
mov [rbp+20h], rcx
mov rax, [rbp+20h]
mov rcx, rax
call _ZNKSt7_cxx1112basic_stringIwSt11char_traitsIwEsIwEE5c_strEv ; std::__cxx11::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::c_str
lea rax, aCactusReadmeTx ; "CaCtUs.ReAdMe.txt"
mov rdx, rax
mov rax, cs: __imp_StrStrIW
call rax ; __imp_StrStrIW
test rax, rax
jnz short loc_140002ACF

mov rax, [rbp+20h]
mov rcx, rax
call _ZNKSt7_cxx1112basic_stringIwSt11char_traitsIwEsIwEE5c_strEv ; std::__cxx11::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::c_str
mov rcx, rax
lea rax, aDesktopIni ; "desktop.ini"
mov rdx, rax
mov rax, cs: __imp_StrStrIW
call rax ; __imp_StrStrIW
test rax, rax
jnz short loc_140002ACF

```

80.00% (113, 128) (601, 414) 00002042 0000000140002A42: checkExt(std::__cxx11::basic_st (Synchronized with Hex View-1)

```

call rax ; __imp_StrStrIW
test rax, rax
jnz short loc_140002ACF

mov rax, [rbp+20h]
mov rcx, rax
call _ZNKSt7_cxx1112basic_stringIwSt11char_traitsIwEsIwEE5c_strEv ; std::__cxx11::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::c_str
mov rcx, rax
lea rax, aUpdateLog ; "update.log"
mov rdx, rax
mov rax, cs: __imp_StrStrIW
call rax ; __imp_StrStrIW
test rax, rax
jnz short loc_140002AD6

mov rax, [rbp+20h]
mov rcx, rax
call _ZNKSt7_cxx1112basic_stringIwSt11char_traitsIwEsIwEE5c_strEv ; std::__cxx11::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::c_str
mov rcx, rax
lea rax, aN ; "n"
mov rdx, rax
mov rax, cs: __imp_StrStrIW
call rax ; __imp_StrStrIW
test rax, rax
jz short loc_140002AD6

```

80.00% (81, 541) (551, 411) 00002042 0000000140002A42: checkExt(std::__cxx11::basic_st (Synchronized with Hex View-1)

L'indirizzo e-mail **cactus[@]mexicomail[.]com** è l'indirizzo di contatto dei malcoders:

```

    text "UTF-16LE", 0Ah,0
    align 20h
F:          ; DATA XREF: makeReadMeFile(std::__cxx11::basic_string<wchar_t,std::char_traits<wchar_t>...
    text "UTF-16LE", 'To recover your files and prevent data disclosure c'
    text "UTF-16LE", 'ontact us via email: cactus@mexicomail.com',0
    align 20h
e:          ; DATA XREF: makeReadMeFile(std::__cxx11::basic_string<wchar_t,std::char_traits<wchar_t>...
    text "UTF-16LE", 'Your unique ID reference: ',0
    align 8
T:          ; DATA XREF: makeReadMeFile(std::__cxx11::basic_string<wchar_t,std::char_traits<wchar_t>...
    text "UTF-16LE", 'Backup contact: TOX (https://tox.chat/):',0
    align 10h
B:          ; DATA XREF: makeReadMeFile(std::__cxx11::basic_string<wchar_t,std::char_traits<wchar_t>...
    text "UTF-16LE", '7367B422CD7498D5F2AAF33F58F67A332F8520CF0279A5FBB46'
    text "UTF-16LE", '11E0121AE421AE1D49ACEABB2',0
    align 10h
S:          ; DATA XREF: makeNtUserFile(std::__cxx11::basic_string<wchar_t,std::char_traits<wchar_t>...
    text "UTF-16LE", '5468652046726565646F6D206F6620496E666F726D6174696F6'
    text "UTF-16LE", 'E204163740A5468652046726565646F6D206F6620496E666F72'
    text "UTF-16LE", '6D6174696F6E204163742028464F4941294F70656E732074686'
    text "UTF-16LE", '973207765627369746520687474703A2F2F7777772E6A757374'
    text "UTF-16LE", '6963652E676F762F6F69702F616D656E6465642D666F69612D07'
    text "UTF-16LE", '265646C696E65642D323031302E70646620696E206E65772077'
    text "UTF-16LE", '696E646F772E2067656E6572616C6C792070726F766964657332'
    text "UTF-16LE", '07468617420616E7920706572736F6E20686173207468652072'
    text "UTF-16LE", '6967687420746F20726571756573742061636365737320746F2'
    00405730|0000000140406730: .rdata:a54686520467265 (Synchronized with Hex View-1)

```

Qui una stringa di introduzione del file di readme del ransomware:

Address	Function	Instruction
.text:00000001400017AF	<u>_Z13cryptFullFileNST7__cxx...</u>	call EVP_EncryptUpdate
.text:0000000140001889	<u>_Z13cryptFullFileNST7__cxx...</u>	call EVP_EncryptUpdate
.text:0000000140001FA5	<u>_Z13cryptPartFileNST7__cxx...</u>	call EVP_EncryptUpdate
.text:00000001400039AB	<u>_Z14makeReadMeFileNST7__...</u>	lea rdx, aYourSystemsWer ; 'Your systems were accessed and encrypte ...'
.text:0000000140013135	aes_cbc_cipher	call EVP_CIPHER_CTX_is_encrypting
.text:0000000140013170	aes_cbc_cipher	call EVP_CIPHER_CTX_is_encrypting
.text:0000000140013196	aes_cbc_cipher	call CRYPTO_cbc128_encrypt
.text:0000000140013215	aes_init_key	call AES_set_encrypt_key
.text:000000014001321C	aes_init_key	mov rcx, cs:_refptr_AES_encrypt
.text:000000014001328C	aes_init_key	lea rdx, asm_AES_cbc_encrypt
.text:00000001400132E9	aes_init_key	call vpaes_set_encrypt_key
.text:00000001400132F0	aes_init_key	mov rcx, cs:_refptr_vpaes_encrypt
.text:0000000140013354	aes_init_key	call AES_set_encrypt_key
.text:0000000140013361	aes_init_key	lea rdx, asm_AES_cbc_encrypt
.text:0000000140013378	aes_init_key	call AES_set_encrypt_key
.text:000000014001337D	aes_init_key	lea rdi, asm_AES_encrypt
.text:0000000140013384	aes_init_key	lea rdx, ossl_bsaes_ctr32_encrypt_blocks
.text:0000000140013380	aes_init_key	call vpaes_set_encrypt_key
.text:00000001400133BD	aes_init_key	lea rdx, vpaes_cbc_encrypt
.text:000000014001340C	aes_init_key	lea rdx, ossl_bsaes_cbc_encrypt
.text:0000000140013446	aesni_cbc_cipher	call EVP_CIPHER_CTX_is_encrypting
.text:0000000140013470	aesni_cbc_cipher	call aesni_cbc_encrypt
.text:00000001400134ED	aesni_init_key	call aesni_set_encrypt_key
.text:00000001400134F2	aesni_init_key	mov rdx, cs:_refptr_aesni_encrypt
.text:0000000140013505	aesni_init_key	mov rdx, cs:_refptr_aesni_cbc_encrypt
.text:0000000140013530	aesni_init_key	call aesni_set_encrypt_key

All'interno della funzione *cryptFullFile* è stata richiamata la funzione *EVP_EncryptUpdate* e funzioni di *istream* e *ostream* per la gestione degli attributi dei files presi in considerazione durante la fase di encryption:

```

call    EVP_EncryptUpdate
mov     rax, [rbp+50250h]
neg     eax
mov     [rbp+50264h], eax
mov     eax, [rbp+50264h]
movsxd rdx, eax
lea     rax, [rbp+50020h]
mov     r8d, 1
mov     rcx, rax
call    _ZN5i5seekExSt12_Ios_Seekdir ; std::istream::seekg(long long,std::_Ios_Seekdir)
mov     eax, [rbp+0]
movsxd r8, eax
lea     rax, [rbp+10h]
lea     rdx, [rbp+50020h]
lea     rcx, [rdx+10h]
mov     rdx, rax ; __int64
call    _ZN5o5writeEPKcx ; std::ostream::write(char const*,long long)
mov     rax, [rbp+50258h]
sub     rax, [rbp+50250h]
mov     rdx, rax
lea     rax, [rbp+50020h]
mov     r8d, 1
mov     rcx, rax
call    _ZN5i5seekExSt12_Ios_Seekdir ; std::istream::seekg(long long,std::_Ios_Seekdir)
mov     rdx, rax
mov     rax, [rbp+50288h]
add     rax, rdx
mov     [rbp+50288h], rax

```

100.00% (355, 3151) (699, 414) 00000DAF 000000001400017AF: cryptFullFile(std::__cxx11::b (Synchronized with Hex View-1)

```

; cryptFullFile(std::__cxx11::basic_string<wchar_t, std::char_traits<wchar_t>, std::allocator<wchar_t>>, unsigned long long, evp_pkey_st *)
public _Z13cryptFullFileNSt7_cxx11i2basic_stringIwSt11char_traitsIwESaIwEEEyP11evp_pkey_st
_Z13cryptFullFileMSt7_cxx11i2basic_stringIwSt11char_traitsIwESaIwEEEyP11evp_pkey_st proc near
var_50328= qword ptr -50328h
var_50320= qword ptr -50320h
var_50318= dword ptr -50318h
var_50308= byte ptr -50308h
var_502C8= byte ptr -502C8h

push   rbp
push   r15
push   r14
push   r13
push   r12
push   rsi
push   rbx
mov     eax, 50310h
call    ___chkstk_ms
sub     rsp, rax
lea     rbp, [rsp+50348h+var_502C8]
mov     [rbp+502D0h], rcx
mov     [rbp+502D8h], rdx
mov     [rbp+502E0h], r8
mov     dword ptr [rbp+50209h], 21217E7Eh
mov     dword ptr [rbp+5020Ch], 217E7E21h
lea     rax, [rbp+50020h]
mov     rcx, rax
call    _ZNSt13basic_fstreamIcSt11char_traitsIcEEC1Ev ; std::basic_fstream<char,std::char_traits<char>>::basic_fstream(void)

```

80.00% (-19, -14) (654, 409) 00000A51 00000000140001451: cryptFullFile(std::__cxx11::bas (Synchronized with Hex View-1)

Inizialmente viene "appesa" l'estensione ".cts0" ai files criptati:


```

mov     dword ptr [rbp+50284h], 28000h
mov     dword ptr [rbp+0], 0
mov     qword ptr [rbp+50278h], 0
call    EVP_CIPHER_CTX_new
mov     [rbp+50278h], rax
mov     rax, [rbp+502E0h]
mov     rcx, rax
call    EVP_PKEY_get_size
cdq     rcx, rax
call    malloc
mov     [rbp+8], rax
mov     ecx, 10h
call    malloc
mov     [rbp+50270h], rax
mov     qword ptr [rbp+50268h], 10h
call    EVP_aes_256_cbc
mov     rcx, rax
lea     r9, [rbp+4]
lea     r8, [rbp+8]
mov     rax, [rbp+50278h]
mov     [rsp+50348h+var_50318], 1
lea     rdx, [rbp+502E0h]
mov     [rsp+50348h+var_50320], rdx
mov     rdx, [rbp+50270h]
mov     [rsp+50348h+var_50328], rdx
mov     rdx, rcx
mov     rcx, rax
call    EVP_SealInit
lea     rax, [rbp-20h]
mov     rdx, [rbp+502D0h]
lea     r8, aCts0 ; ".cts0"
mov     rcx, rax
call    _ZStplIwSt11char_traitsIwESaIwEENSt7__cxx112basic_stringIT_T0_T1_EERKS8_PKSS_ ; std::operator+<wchar_t,std::char_traits<wchar_t>,std:
lea     rax, [rbp-40h]

```

80.00% (-16,502) | (€19,398) | 00000B28 | 0000000140001528: cryptFullFile(std::__cxx11::bas | (Synchronized with Hex View-1)

Successivamente viene sostituita con l'estensione ".cts1":

```

mov     rdx, rcx
mov     rcx, rax
call    EVP_SealInit
lea     rax, [rbp-20h]
mov     rdx, [rbp+502D0h]
lea     r8, aCts0 ; ".cts0"
mov     rcx, rax
call    _ZStplIwSt11char_traitsIwESaIwEENSt7__cxx112basic_stringIT_T0_T1_EERKS8_PKSS_ ; std::operator+<wchar_t,std::char_traits<wchar_t>,std:
lea     rax, [rbp-40h]
mov     rdx, [rbp+502D0h]
lea     r8, aCts1 ; ".cts1"
mov     rcx, rax
call    _ZStplIwSt11char_traitsIwESaIwEENSt7__cxx112basic_stringIT_T0_T1_EERKS8_PKSS_ ; std::operator+<wchar_t,std::char_traits<wchar_t>,std:
lea     rax, [rbp-20h]
mov     rcx, rax
call    _ZNKSt7__cxx112basic_stringIwSt11char_traitsIwESaIwEE5c_strEv ; std::__cxx11::basic_string<wchar_t,std::char_traits<wchar_t>,std::all
mov     rbx, rax
mov     rax, [rbp+502D0h]
mov     rcx, rax
call    _ZNKSt7__cxx112basic_stringIwSt11char_traitsIwESaIwEE5c_strEv ; std::__cxx11::basic_string<wchar_t,std::char_traits<wchar_t>,std::all
mov     rdx, rbx
mov     rcx, rax
mov     rax, cs: __imp_wrename
call    rax ; __imp_wrename
mov     edx, 8
mov     ecx, 10h
call    _ZStorSt13_Ios_OpenmodeS_ ; std::operator|(std::_Ios_Openmode,std::_Ios_Openmode)
mov     edx, 4
mov     ecx, eax
call    _ZStorSt13_Ios_OpenmodeS_ ; std::operator|(std::_Ios_Openmode,std::_Ios_Openmode)
mov     ebx, eax
lea     rax, [rbp-20h]
mov     rcx, rax
call    _ZNKSt7__cxx112basic_stringIwSt11char_traitsIwESaIwEE5c_strEv ; std::__cxx11::basic_string<wchar_t,std::char_traits<wchar_t>,std::all
mov     rdx, rax

```

80.00% (-16,896) | (€54,391) | 00000B9D | 000000014000159D: cryptFullFile(std::__cxx11::bas | (Synchronized with Hex View-1)

Vi è l'utilizzo della wildcard "*" per la gestione di filesystem enumeration:

```

add     rax, 008h
mov     rcx, rax
call   _ZNKSt9basic_iosIcSt11char_traitsIcEE3eofEv ; std::basic_ios<char,std::char_traits<char>>::eof(void)
xor     eax, 1
test    al, al
jnz    loc_1400016C4

loc_1400016C4:
mov     rax, [rbp+50288h]
mov     rdx, [rbp+50208h]
sub     rdx, rax
mov     [rbp+50258h], rdx
mov     eax, [rbp+50284h]
cdq
cmp     [rbp+50258h], rax
jge    loc_140001841

loc_140001841:
mov     eax, [rbp+50284h]
movsxd rcx, eax
lea     rdx, [rbp+28020h] ; _
lea     rax, [rbp+50020h]
mov     r8, rcx
mov     rcx, rax
call   _ZNSi4readEPcx ; std
lea     r8, [rbp+28020h]
mov     rcx, rbp

```

80.00% (409, 2166) (579, 396) 000000B9D 0000000014000159D: cryptFullFile(std::__cxx11::ba (Synchronized with Hex View-1)

A seguire un riferimento ad un oggetto di encryption key contestualmente al richiamo della funzione cryptFullFile:

```

call   rax ; __imp_wrename
lea    rax, [rbp-40h]
mov    rcx, rax
call   _ZNSt7_cxx112basic_stringIwSt11char_traitsIwESaIwEEED1Ev ; std::__cxx11::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator
lea    rax, [rbp-20h]
mov    rcx, rax
call   _ZNSt7_cxx112basic_stringIwSt11char_traitsIwESaIwEEED1Ev ; std::__cxx11::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator
lea    rax, [rbp+50020h]
mov    rcx, rax
call   _ZNSt13basic_fstreamIcSt11char_traitsIcEEED1Ev ; std::basic_fstream<char,std::char_traits<char>>::~basic_fstream()
jmp    short loc_140001AF7

loc_140001AF7:
lea    rsp, [rbp+50290h]
pop    rbx
pop    rsi
pop    r12
pop    r13
pop    r14
pop    r15
pop    rbp
retn
_Z13cryptFullFileNSt7_cxx112basic_stringIwSt11char_traitsIwESaIwEEE_P11evp_pkey_st_endp

```

80.00% (161, 5048) (696, 418) 000000A50 00000000140001450: cryptFullFile(std::__cxx11::ba (Synchronized with Hex View-1)

Qui la funzione di enumerazione dei dischi della macchina compromessa:

```

; getDrives(std::_cxx11::basic_string<wchar_t, std::char_traits<wchar_t>, std::allocator<wchar_t>> *)
public _Z9getDrivesPNSt7__cxx112basic_stringIwSt11char_traitsIwESaIwEEEE
_Z9getDrivesPNSt7__cxx112basic_stringIwSt11char_traitsIwESaIwEEEE proc near

var_1C= dword ptr -1Ch
var_18= qword ptr -18h
var_10= qword ptr -10h
var_4= dword ptr -4
arg_0= qword ptr 10h

push rbp
mov rbp, rsp
sub rsp, 40h
mov [rbp+arg_0], rcx
mov [rbp+var_4], 0
mov ecx, 400h
call malloc
mov [rbp+var_18], rax
mov rax, [rbp+var_18]
mov rdx, rax
mov ecx, 400h
mov rax, cs:__imp_GetLogicalDriveStringsw
call rax ; __imp_GetLogicalDriveStringsw
mov rax, [rbp+var_18]
mov [rbp+var_10], rax
jmp short loc_1400022C4

```

100.00% (424, 79) (491, 406) 00001830 0000000140002230: getDrives(std::_cxx11::basic_s (Synchronized with Hex View-1)

```

; checkProcess(std::_cxx11::basic_string<wchar_t, std::char_traits<wchar_t>, std::allocator<wchar_t>>)
public _Z12checkProcessNSt7__cxx112basic_stringIwSt11char_traitsIwESaIwEEEE
_Z12checkProcessNSt7__cxx112basic_stringIwSt11char_traitsIwESaIwEEEE proc near

var_18= byte ptr -18h

push rbp
push rbx
sub rsp, 38h
lea rbp, [rsp+48h+var_18]
mov [rbp+20h], rcx
mov dword ptr [rbp-8], 0Eh
mov dword ptr [rbp-4], 0
jmp short loc_14000234C

```

```

loc_14000234C:
mov eax, [rbp-4]
cmp eax, [rbp-8]
jl short loc_140002304

```

80.00% (8, -36) (298, 418) 000018E5 00000001400022E5: checkProcess(std::_cxx11::basic (Synchronized with Hex View-1)

A seguire i dettagli della funzione *killFileProcesses* che include la gestione del *RestartManager* per "liberare" le risorse dei files da criptare dai processi che li stanno (eventualmente) utilizzando.

```

;__int64 __fastcall killFileProcesses(const wchar_t *)
public _Z17killFileProcessesPKw
_Z17killFileProcessesPKw proc near

var_348= qword ptr -348h
var_340= dword ptr -340h
var_338= qword ptr -338h
var_2E8= byte ptr -2E8h

push    rbp
push    rsi
push    rbx
sub     rsp, 350h
lea     rbp, [rsp+368h+var_2E8]
mov     [rbp+2F0h], rcx
mov     byte ptr [rbp+2CFh], 0
mov     dword ptr [rbp+278h], 0
mov     dword ptr [rbp+2C4h], 0
pxor    xmm0, xmm0
movups  xmmword ptr [rbp-40h], xmm0
movdqu  xmm1, xmmword ptr [rbp-40h]
movups  xmmword ptr [rbp+230h], xmm1
movdqu  xmm2, xmmword ptr [rbp-40h]
movups  xmmword ptr [rbp+240h], xmm2
movdqu  xmm3, xmmword ptr [rbp-40h]

```

100.00% (2878, 21) (543, 420) 00001960 0000000140002360: killFileProcesses(wchar_t cons (Synchronized with Hex View-1)

```

mov     [rsp+368h+var_348], 0
mov     r9d, 0 ; nServices
lea     r8, [rbp+2F0h] ; rgApplications
mov     edx, 1 ; rgsFileNames
mov     ecx, eax ; nApplications
call    RmRegisterResources
test    eax, eax
setz   al
test    al, al
jz     loc_140002838

```

```

mov     dword ptr [rbp+22Ch], 0
mov     dword ptr [rbp+228h], 0
mov     dword ptr [rbp+224h], 0
mov     qword ptr [rbp+288h], 0
mov     eax, [rbp+278h]
lea     r8, [rbp+224h] ; lpdwRebootReasons
lea     rdx, [rbp+228h] ; pnProcInfo
lea     rcx, [rbp+22Ch]
mov     [rsp+368h+var_348], rcx
mov     r9d, 0
mov     ecx, eax ; rgAffectedApps
call    RmGetList
mov     [rbp+2C4h], eax
cmp     dword ptr [rbp+2C4h], 0EAh

```

100.00% (2887, 771) (668, 394) 00001A3E 000000014000243E: killFileProcesses(wchar_t con (Synchronized with Hex View-1)

```

mov     rax, [rbp+288h]
add     rax, rdx
mov     eax, [rax]
mov     r8d, eax
mov     edx, 0 ; dwProcessId
mov     ecx, 1FFFFFFh
mov     rax, cs:__imp_OpenProcess
call    rax ; __imp_OpenProcess
mov     [rbp+2A8h], rax
cmp     qword ptr [rbp+2A8h], 0
jz     loc_1400027D6

```

```

lea     rdx, [rbp+10h] ; nSize
mov     rax, [rbp+2A8h]
mov     r8d, 104h
mov     rcx, rax
call    K32GetProcessImageFileNameW
mov     rax, [rbp+2A8h]
mov     rcx, rax
mov     rax, cs:__imp_CloseHandle
call    rax ; __imp_CloseHandle
lea     rax, [rbp+27Fh]
mov     rcx, rax
call    _ZNSaIwEC1Ev ; std::allocator<wchar_t>::allocator(void)
lea     rcx, [rbp+27Fh]

```

100.00% (3428, 2664) (799, 405) 00001A3E 000000014000243E: killFileProcesses(wchar_t co (Synchronized with Hex View-1)

Di seguito un riferimento che sembrerebbe essere associato all'accesso a network shares, nonché un accesso concorrente ai files da criptare, prendendo in considerazione gli attributi degli stessi ed attraverso una conferma booleana:

```

lea     rdx, [rbp+10h]
lea     rax, [rbp-30h]
mov     r8, rcx
mov     rcx, rax
call   _ZNSt7__cxx1112basic_stringIwSt11char_traitsIwESaIwEEC1IS3_EEPKwRKS3_ ; std::__cxx11::basic_string<wchar
lea     rax, [rbp+27Fh]
mov     rcx, rax
call   _ZNSaIwED1Ev ; std::allocator<wchar_t>::~allocator()
lea     rax, [rbp-30h]
mov     r8, 0FFFFFFFFFFFFFFFh
lea     rdx, asc_140406392 ; "\\\"
mov     rcx, rax
call   _ZNKSt7__cxx1112basic_stringIwSt11char_traitsIwESaIwEE12find_last_ofEPKwy ; std::__cxx11::basic_string<w
mov     [rbp+2A4h], eax
mov     eax, [rbp+2A4h]
add     eax, 1
movsxd rcx, eax
lea     rax, [rbp-10h]
lea     rdx, [rbp-30h]
mov     r9, 0FFFFFFFFFFFFFFFh
mov     r8, rcx
mov     rcx, rax
call   _ZNKSt7__cxx1112basic_stringIwSt11char_traitsIwESaIwEE6substrEyy ; std::__cxx11::basic_string<wchar_t,st
lea     rdx, [rbp-10h]
lea     rax, [rbp+280h]
mov     rcx, rax
call   _ZNSt7__cxx1112basic_stringIwSt11char_traitsIwESaIwEEC1ERKS4_ ; std::__cxx11::basic_string<wchar_t,std::
lea     rax, [rbp+280h]

```

100.00% | (3428, 3084) | (716, 338) | 00001A3E | 0000000140002436: killFileProcesses(wchar_t co) (Synchronized with Hex View-1)

```

public _Z10IsFileFreePKw
_Z10IsFileFreePKw proc near
var_40= dword ptr -40h
var_38= dword ptr -38h
var_30= qword ptr -30h
var_14= dword ptr -14h
var_10= qword ptr -10h
var_1= byte ptr -1
lpSecurityAttributes= qword ptr 10h
push   rbp
mov     rbp, rsp
sub     rsp, 60h
mov     [rbp+lpSecurityAttributes], rcx
mov     [rbp+var_1], 1
mov     rax, [rbp+lpSecurityAttributes]
mov     [rsp+60h+var_30], 0
mov     [rsp+60h+var_38], 0
mov     [rsp+60h+var_40], 3
mov     r9d, 0 ; dwFlagsAndAttributes
mov     r8d, 0 ; dwCreationDisposition
mov     edx, 0C0000000h ; dwShareMode
mov     rcx, rax ; lpSecurityAttributes
mov     rax, cs:__imp_CreateFileW
call   rax ; __imp_CreateFileW
mov     [rbp+var_10], rax
mov     rax, cs:__imp_GetLastError

```

100.00% | (-152, 96) | (409, 420) | 00001F35 | 0000000140002935: IsFileFree(wchar_t const*) (Synchronized with Hex View-1)

```

jz short loc_1400029A2

cmp [rbp+var_14], 21h ; '!
jnz short loc_140002A03

loc_1400029A2:
mov rax, [rbp+lpSecurityAttributes]
mov rcx, rax
call _Z17killFileProcessesPKw ; killFileProcesses(wchar_t const*)
test al, al
jz short loc_1400029FD

mov rax, [rbp+lpSecurityAttributes]
mov [rsp+60h+var_30], 0
mov [rsp+60h+var_38], 0
mov [rsp+60h+var_40], 3
mov r9d, 0 ; dwFlagsAndAttributes
mov r8d, 0 ; dwCreationDisposition
mov edx, 0C0000000h ; dwShareMode
  
```

100.00% (-109, 656) (746, 400) 00001F35 0000000140002935: IsFileFree(wchar_t const*) (Synchronized with Hex View-1)

Lo script *rn.bat* viene molto probabilmente utilizzato in fase di rinomina dei files presi in considerazione durante l'infection kill chain:

```

mov r8, rcx
mov rcx, rax
call _ZN17_cxx112basic_stringISt11char_traitsIwESt1wESt13_FEPwAKS3_ ; std::__cxx11::basic_stringochar_t, std::char_traitsochar_t, std::allocatorochar_t>::bas
lea rax, [rbp+380h]
mov rcx, rax
call _ZN16wE1E ; std::allocatorochar_t::allocator()
mov rax, [rbp+400h]
mov r8, rax
lea rdx, asc_140406392 ; '\\
mov rcx, rax
call _ZN17_cxx112basic_stringISt11char_traitsIwESt1wESt13_FEPw ; std::__cxx11::basic_stringochar_t, std::char_traitsochar_t, std::allocatorochar_t>::
lea rax, [rbp+390h]
mov rcx, rax
mov rdx, [rbp+400h]
mov r8, rcx
mov r8d, 0
mov rcx, rax
call _ZN17_cxx112basic_stringISt11char_traitsIwESt1wESt13_FEPw ; std::__cxx11::basic_stringochar_t, std::char_traitsochar_t, std::allocatorochar_t>::substr(ul
lea rdx, [rbp+390h]
mov rcx, rax
mov rdx, [rbp+400h]
call _ZN17_cxx112basic_stringISt11char_traitsIwESt1wESt13_FEPw ; std::__cxx11::basic_stringochar_t, std::char_traitsochar_t, std::allocatorochar_t>::operator=(std
lea rax, [rbp+390h]
mov rcx, rax
call _ZN17_cxx112basic_stringISt11char_traitsIwESt1wESt13_FEPw ; std::__cxx11::basic_stringochar_t, std::char_traitsochar_t, std::allocatorochar_t>::basic_string()
lea rax, [rbp+380h]
mov rdx, [rbp+400h]
lea r8, [rbp+420h]
mov rcx, rax
call _Z58Bat ; '\\rn.bat'
lea rax, [rbp+380h]
call _Z58Bat ; '\\rn.bat'
lea rdx, [rbp+380h]
mov rcx, rax
mov rdx, [rbp+400h]
call _ZN17_cxx112basic_stringISt11char_traitsIwESt1wESt13_FEPw ; std::__cxx11::basic_stringochar_t, std::char_traitsochar_t, std::allocatorochar_t>::operator=(std
lea rax, [rbp+380h]
mov rcx, rax
call _ZN17_cxx112basic_stringISt11char_traitsIwESt1wESt13_FEPw ; std::__cxx11::basic_stringochar_t, std::char_traitsochar_t, std::allocatorochar_t>::basic_string()
jmp short loc_140003089
  
```

64.00% (-69, 433) (489, 417) 00002533 0000000140002F33: extractBatPath(void) (Synchronized with Hex View-1)

La funzione *extractDirPath* ottiene i filenames dato un path specifico:

```

; extractDirPath[abi:cxx11](void)
public _Z14extractDirPathB5cxx1lv
_Z14extractDirPathB5cxx1lv proc near

var_3D8= byte ptr -3D8h

push    rbp
push    rbx
sub     rsp, 448h
lea    rbp, [rsp+458h+var_3D8]
mov    [rbp+3E0h], rcx
lea    rax, [rbp-60h]
mov    r8d, 1F4h
mov    rdx, rax          ; nSize
mov    ecx, 0
mov    rax, cs: __imp_GetModuleFileNameW
call   rax ; __imp_GetModuleFileNameW
mov    [rbp+3BCh], eax
lea    rax, [rbp+38Fh]
mov    rcx, rax
call   _ZNSt7__cxx11::basic_stringIwSt11char_traitsIwESaIwEEC1IS3_EEPKwRKKS3_ ; std::allocator<wchar_t>::allocator(void)
lea    rcx, [rbp+38Fh]
lea    rdx, [rbp-60h]
mov    rax, [rbp+3E0h]
mov    r8, rcx
mov    rcx, rax
call   _ZNSt7__cxx11::basic_stringIwSt11char_traitsIwESaIwEE12find_last_ofEPKw ; std::__cxx11::basic_string<wchar_t,std::char_traits<wchar_t>
lea    rax, [rbp+38Fh]
mov    rcx, rax
call   _ZNSt7__cxx11::basic_stringIwSt11char_traitsIwESaIwEE10atow ; std::allocator<wchar_t>::~~allocator()
mov    rax, [rbp+3E0h]
mov    r8, 0FFFFFFFFFFFFFFFh
lea    rdx, asc_140406392 ; "\\\"
mov    rcx, rax

```

80.00% | (-76, 50) | (297, 398) | 0000273E | 000000014000313E: extractDirPath(void) | (Synchronized with Hex View-1)

```

mov    rax, [rbp+3E0h]
mov    r8, 0FFFFFFFFFFFFFFFh
lea    rdx, asc_140406392 ; "\\\"
mov    rcx, rax
call   _ZNKSt7__cxx11::basic_stringIwSt11char_traitsIwESaIwEE12find_last_ofEPKw ; std::__cxx11::basic_string<wchar_t,std::char_traits<wchar_t>
mov    rcx, rax
lea    rax, [rbp+390h]
mov    rdx, [rbp+3E0h]
mov    r9, rcx
mov    r8d, 0
mov    rcx, rax
call   _ZNKSt7__cxx11::basic_stringIwSt11char_traitsIwESaIwEE6substrEyy ; std::__cxx11::basic_string<wchar_t,std::char_traits<wchar_t>,s
lea    rdx, [rbp+390h]
mov    rax, [rbp+3E0h]
mov    rcx, rax
call   _ZNSt7__cxx11::basic_stringIwSt11char_traitsIwESaIwEEaSE054_ ; std::__cxx11::basic_string<wchar_t,std::char_traits<wchar_t>,std::
lea    rax, [rbp+390h]
mov    rcx, rax
call   _ZNSt7__cxx11::basic_stringIwSt11char_traitsIwESaIwEE10atow ; std::__cxx11::basic_string<wchar_t,std::char_traits<wchar_t>,std::all
jmp     short loc_140003252

```

loc_140003252:

```

mov    rax, [rbp+3E0h]
add    rsp, 448h
pop    rbx
pop    rbp
retn
_Z14extractDirPathB5cxx1

```

80.00% | (-54, 513) | (580, 398) | 0000273E | 000000014000313E: extractDirPath(void) | (Synchronized with Hex View-1)

Cactus Ransomware possiede un attributo globale, il quale sembrerebbe far riferimento ad una blacklist di estensioni, difatti denominata *extBlackList*:

```

; Attributes: bp-based frame

_GLOBAL__sub_I_extBlackList proc near
push    rbp
mov     rbp, rsp
sub     rsp, 20h
mov     edx, 0FFFFh
mov     ecx, 1
call   _Z41__static_initialization_and_destruction_0ii ; __static_initialization_and_destruction_0(int,int)
nop
add     rsp, 20h
pop     rbp
retn
_GLOBAL__sub_I_extBlackList endp

```

100.00% (-25, -70) (586, 418) 000028C3 | 00000001400032C3: _GLOBAL__sub_I_extBlackList (Synchronized with Hex View-1)

La funzione *hexEncode* prende come parametro in input la variabile *var_178*, utilizzata in un'istruzione *lea* con il registro *rbp*, per caricare l'indirizzo di memoria in questione:

```

; hexEncode(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char>> *, std::__cxx11::basic_string<wchar_t, std::char_traits<wchar_t>, std::allocator<wchar_t>> *)
public _Z9hexEncodePNSt7__cxx112basic_stringIcSt11char_traitsIcESaIcEEENS0_IwS1_IwESaIwEEEE
_Z9hexEncodePNSt7__cxx112basic_stringIcSt11char_traitsIcESaIcEEENS0_IwS1_IwESaIwEEEE proc near
var_178= byte ptr -178h
push    rbp
push    rbx
sub     rsp, 1E8h
lea     rbp, [rsp+1F8h+var_178]
mov     [rbp+180h], rcx
mov     [rbp+188h], rdx
mov     rax, [rbp+180h]
mov     rcx, rax
call   _ZNKSt7__cxx112basic_stringIwSt11char_traitsIwESaIwEE6lengthEv ; std::__cxx11::basic_string<wchar_t, std::char_traits<wchar_t>, std::allocator<wchar_t>>.length() const
mov     [rbp+150h], rax
lea     rax, [rbp-60h]
mov     rcx, rax
call   _ZNSt7__cxx1118basic_stringstreamIcSt11char_traitsIcESaIcEEC1Ev ; std::__cxx11::basic_stringstream<char, std::char_traits<char>, std::allocator<char>>.basic_stringstream(char const*)
mov     qword ptr [rbp+158h], 0
jmp     loc_1400033C6

```

80.00% (393, -31) (479, 406) 000028F0 | 00000001400032F0: hexEncode(std::__cxx11::basic_s (Synchronized with Hex View-1)

Qui la wildcard "*", utilizzata in fase di files enumeration loop per ottenere i files da sottoporre al processo di crittografia:

```

mov rbx, rax

; SearchFiles(std::__cxx11::basic_string<wchar_t, std::char_traits<wchar_t>, std::allocator<wchar_t>>)
public _Z11SearchFilesNSt7__cxx112basic_stringIwSt11char_traitsIwESaIwEEEE
_Z11SearchFilesNSt7__cxx112basic_stringIwSt11char_traitsIwESaIwEEEE proc near

var_588= byte ptr -588h

push rbp
push rbx
sub rsp, 5F8h
lea rbp, [rsp+600h+var_588]
mov [rbp+590h], rcx
lea rax, [rbp+20h]
mov rdx, [rbp+50h]
lea r8, asc_14040C58 ; "\\*"
mov rcx, rax
call _Z5tplIwSt11char_traitsIwESaIwEENSt7__cxx112basic_stringIT_T0_T1_EERKS8_PK55_ ; std::operator+<wchar_t,std::cha
lea rax, [rbp+20h]
mov rcx, rax
call _Z9NSt7__cxx112basic_stringIwSt11char_traitsIwESaIwEE5c_strEv ; std::__cxx11::basic_string<wchar_t,std::char_tr
mov rcx, rax
lea rax, [rbp+40h]
mov rdx, rax
mov rax, cs: _imp_FindFirstFileW
call rax ; _imp_FindFirstFileW
mov [rbp+560h], rax
lea rax, [rbp+290h]
mov rcx, rax
call _Z9SaIwEC1Ev ; std::allocator<wchar_t>::allocator(void)
lea rdx, [rbp+290h]
mov rax, rbp

```

80.00% (18767, -1) (582, 415) 000035E2 0000000140003F62: SearchFiles(std::__cxx11::basi (Synchronized with Hex View-1)

```

call _Z9NSt7__cxx112basic_stringIwSt11char_traitsIwESaIwEEC1ERKS4_ ; std::__cxx11::basic_string<wchar_t,std::char_traits<wchar_t>,s
lea rax, [rbp+3F0h]
mov rdx, rbp
lea r8, asc_14040C6A ; "\""
mov rcx, rax
call _Z5tplIwSt11char_traitsIwESaIwEENSt7__cxx112basic_stringIT_T0_T1_EERKS8_PK55_ ; std::operator+<wchar_t,std::char_traits<wchar
lea rax, [rbp+3D0h]
lea rdx, [rbp+3F0h]
lea r8, _Z13txtReadMeNameB5cxx11 ; txtReadMeName
mov rcx, rax
call _Z5tplIwSt11char_traitsIwESaIwEENSt7__cxx112basic_stringIT_T0_T1_EE058_RKS8_ ; std::operator+<wchar_t,std::char_traits<wchar_t
lea rdx, [rbp+380h]
lea rax, [rbp+3D0h]
mov rcx, rax
call _Z14makeReadMeFileNSt7__cxx112basic_stringIwSt11char_traitsIwESaIwEEES4_ ; makeReadMeFile(std::__cxx11::basic_string<wchar_t,
lea rax, [rbp+3D0h]
mov rcx, rax
call _Z9NSt7__cxx112basic_stringIwSt11char_traitsIwESaIwEED1Ev ; std::__cxx11::basic_string<wchar_t,std::char_traits<wchar_t>,std::
lea rax, [rbp+3F0h]
mov rcx, rax
call _Z9NSt7__cxx112basic_stringIwSt11char_traitsIwESaIwEED1Ev ; std::__cxx11::basic_string<wchar_t,std::char_traits<wchar_t>,std::
lea rax, [rbp+380h]
mov rcx, rax
call _Z9NSt7__cxx112basic_stringIwSt11char_traitsIwESaIwEED1Ev ; std::__cxx11::basic_string<wchar_t,std::char_traits<wchar_t>,std::
mov rdx, rbp
lea rax, [rbp+410h]
mov rcx, rax
call _Z9NSt7__cxx112basic_stringIwSt11char_traitsIwESaIwEEC1ERKS4_ ; std::__cxx11::basic_string<wchar_t,std::char_traits<wchar_t>,s
lea rax, [rbp+410h]
mov rcx, rax
call _Z11SearchFilesNSt7__cxx112basic_stringIwSt11char_traitsIwESaIwEEEE ; SearchFiles(std::__cxx11::basic_string<wchar_t,std::char
lea rax, [rbp+410h]
mov rcx, rax
call _Z9NSt7__cxx112basic_stringIwSt11char_traitsIwESaIwEED1Ev ; std::__cxx11::basic_string<wchar_t,std::char_traits<wchar_t>,std::
lea rax, [rbp+410h]

```

80.00% (1987, 5702) (254, 403) 000035E2 0000000140003F62: SearchFiles(std::__cxx11::bas (Synchronized with Hex View-1)

I files presi in considerazione vengono salvati in una variabile di *filesArray*, nel caso in cui vi siano errori di accesso agli stessi, viene prodotta una stringa di errore:

```

mov     rax, cs:newArrayFile
shl     rax, 5
mov     rdx, rax
lea     rax, _2109filesArrayB5cx11 ; filesArray
lea     rcx, [rdx+rax]
lea     rax, [rbp-20h]
mov     rdx, rax
call   _20547_cxx112basic_stringI2St11char_traitsIwE5aIwE4S5ERK54 ; std::_cxx11::basic_stringowchar_t,std:
mov     rax, cs:newArrayFile
add     rax, 1
mov     rcx,newArrayFile, rax
jmp     loc_140004770

```

```

rax, [rbp+450h]
rdi, rbp
rsi, rdx
rdi, a5_0 ; "s"
rcx, rax
_210912St11char_traitsIwE5aIwE4S5ERK57_cxx112basic_stringIT_0_T1_EEPK55_NK58 ; std::operator+owchar_t,std:char_traitsowchar_t,std:allocatorowchar_t>>(wchar_t const*,std:
rcx, rax
_201oggerRG57_cxx112basic_stringI2St11char_traitsIwE5aIwE4EE ; logger(std::_cxx11::basic_stringowchar_t,std:char_traitsowchar_t,std:allocatorowchar_t>>
rax, [rbp+450h]
rcx, rax
_20547_cxx112basic_stringI2St11char_traitsIwE5aIwE4EEv ; std::_cxx11::basic_stringowchar_t,std:char_traitsowchar_t,std:allocatorowchar_t>>::basic_string()
loc_140004790

```

64.00% (6423,5254) (515,415) 0000353F 0000000140003F3F: SearchFiles(std::_cxx11::bas (Synchronized with Hex View-1)

```

mov     rcx, cs:logger
lea     rax, [rbp+320h]
mov     rcx, rax
call   _20547_cxx112basic_stringI2St11char_traitsIwE5aIwE4EEv ; std::_cxx11::basic_stringowchar_t,std:

```

```

loc_140004718:
movzx  ebx, cs:needLogger
test   al, al
jz     short loc_140004750

```

```

loc_1400041D0:
lea     rax, [rbp+36Fh]
mov     rcx, rax
call   _2054a4E1Ev ; std:allocatorowchar_t::allocator(void)
lea     rcx, [rbp+36Fh]
lea     rdx, [rbp+40h]
lea     rdx, [rbp+20h]
lea     rax, [rbp+340h]
mov     r8, rcx
mov     rcx, rax
call   _20547_cxx112basic_stringI2St11char_traitsIwE5aIwE4EEv ; std::_cxx11::basic
lea     rax, [rbp+340h]
mov     rcx, rax
call   _214checkFolderEvRG57_cxx112basic_stringI2St11char_traitsIwE5aIwE4EE ; checkFolderExt(std
lea     rax, [rbp+340h]
mov     rcx, rax
call   _20547_cxx112basic_stringI2St11char_traitsIwE5aIwE4EEv ; std::_cxx11::basic_stringowchar
lea     rax, [rbp+36Fh]
mov     rcx, rax
call   _2054a4E1Ev ; std:allocatorowchar_t::allocator()
test   si, si
jnz     loc_1400041B0

```

```

lea     rax, [rbp+500h]
lea     rdx, [rbp+60h]
lea     rax, [rbp+500h]
call   _201oggerRG57_cxx112basic_stringI2St11char_traitsIwE5aIwE4EEv ; logger(std::_cxx11::basic_stringowchar_t,std:char_traitsowchar_t,std:allocatorowchar_t>>
mov     rcx, rax
call   _201oggerRG57_cxx112basic_stringI2St11char_traitsIwE5aIwE4EEv ; logger(std::_cxx11::basic_stringowchar_t,std:char_traitsowchar_t,std:allocatorowchar_t>>
mov     rcx, rax
call   _20547_cxx112basic_stringI2St11char_traitsIwE5aIwE4EEv ; std::_cxx11::basic_stringowchar_t,std:char_traitsowchar_t,std:allocatorowchar_t>>::basic_string()

```

51.20% (10201,4350) (650,407) 0000353F 0000000140003F3F: SearchFiles(std::_cxx11::ba (Synchronized with Hex View-1)

```

call _ZN57_cxx112basic_stringIwSt11char_traitsIwESaIwEEC1ERKSA_ ; std::__cxx11::basic_string<wchar_t, std::char_traits<wchar_t>, std::allocator<wchar_t>>::basic_string(std::__cxx11::basic_string<wchar_t, std::char_traits<wchar_t>, std::allocator<wchar_t>> const&, std::allocator<wchar_t> const&)
lea rax, [rbp+210h]
mov rdx, [rbp+200h]
mov rcx, rax
call _Z13longPathCheckK057_cxx112basic_stringIwSt11char_traitsIwESaIwEEE ; longPathCheck(std::__cxx11::basic_string<wchar_t, std::char_traits<wchar_t>, std::allocator<wchar_t>> const&)
lea rax, [rbp+2E0h]
mov rcx, rax
call _ZN57_cxx112basic_stringIwSt11char_traitsIwESaIwEES0_ ; std::__cxx11::basic_string<wchar_t, std::char_traits<wchar_t>, std::allocator<wchar_t>>::operator=(std::__cxx11::basic_string<wchar_t, std::char_traits<wchar_t>, std::allocator<wchar_t>> const&)
lea rax, [rbp+2E0h]
mov rcx, rax
call _ZN57_cxx112basic_stringIwSt11char_traitsIwESaIwEEDiv ; std::__cxx11::basic_string<wchar_t, std::char_traits<wchar_t>, std::allocator<wchar_t>>::basic_string()
mov rcx, rax
call _ZN57_cxx112basic_stringIwSt11char_traitsIwESaIwEEDiv ; std::__cxx11::basic_string<wchar_t, std::char_traits<wchar_t>, std::allocator<wchar_t>>::basic_string()

loc_1400041B8:
movzx ebx, cs:needExtraLogger
test al, al
jc short loc_1400041D0

lea rax, [rbp+320h]
mov rdx, rbp
mov rbx, rdx
lea rdx, wCheckingFolder ; "checking folder"
mov rcx, rax
call _Z15getInSt11char_traitsIwESaIwEENS7_cxx112basic_stringIT_7T_1_EEPKSS_RKSB_ ; std::operator+<wchar_t, std::char_traits<wchar_t>, std::allocator<wchar_t>>(wchar_t const&, std::__cxx11::basic_string<wchar_t, std::char_traits<wchar_t>, std::allocator<wchar_t>> const&)
lea rax, [rbp+320h]
mov rcx, rax
call _Z16loggerMS7_cxx112basic_stringIwSt11char_traitsIwESaIwEEE ; logger(std::__cxx11::basic_string<wchar_t, std::char_traits<wchar_t>, std::allocator<wchar_t>> const&)
lea rax, [rbp+320h]
mov rcx, rax
call _ZN57_cxx112basic_stringIwSt11char_traitsIwESaIwEEDiv ; std::__cxx11::basic_string<wchar_t, std::char_traits<wchar_t>, std::allocator<wchar_t>>::basic_string()

loc_140004718:
movzx ebx, cs:needLogger
test al, al
jc short loc_140004738

loc_1400041D0:
lea rax, [rbp+36Fh]
mov rcx, rax
call _ZNS4w6C1Ev ; std::allocator<wchar_t>::allocator(void)
lea rcx, [rbp+36Fh]
lea rax, [rbp+40h]
lea rdx, [rax+0Ch]
lea rax, [rbp+340h]
mov rbx, rcx

```

51.20% (10824, 3793) (757, 412) 0000353F 0000000140003F3F: SearchFiles(std::__cxx11::ba (Synchronized with Hex View-1)

Qui un dettaglio di logging di infezione di un disco:

```

lea rax, [rbp-60h]
mov edx, [rbp+104Ch]
movsxd rdx, edx
shl rdx, 5
add rdx, rax
lea rax, [rbp+1000h]
mov r8, rdx
lea rdx, aProcessingDrive ; "processing drive"
mov rcx, rax
call _Z54p1w6St11char_traitsIwESaIwEENS7_cxx112basic_stringIT_7T_1_EEPKSS_RKSB_ ; std::operator+<wchar_t, std::char_traits<wchar_t>, std::allocator<wchar_t>>(wchar_t const&, std::allocator<wchar_t> const&)
lea rax, [rbp+1000h]
mov rcx, rax
call _Z16loggerMS7_cxx112basic_stringIwSt11char_traitsIwESaIwEEE ; logger(std::__cxx11::basic_string<wchar_t, std::char_traits<wchar_t>, std::allocator<wchar_t>> const&)
lea rax, [rbp+1000h]
mov rcx, rax
call _ZN57_cxx112basic_stringIwSt11char_traitsIwESaIwEEDiv ; std::__cxx11::basic_string<wchar_t, std::char_traits<wchar_t>, std::allocator<wchar_t>>::basic_string()

loc_1400048AD:
lea rax, [rbp-60h]
mov edx, [rbp+104Ch]
movsxd rdx, edx
shl rdx, 5
add rdx, rax
lea rax, [rbp+1020h]
mov r8d, 2
mov r8b, 0
mov rcx, rax
call _ZN57_cxx112basic_stringIwSt11char_traitsIwESaIwEESubstrFyy ; std::__cxx11::basic_string<wchar_t, std::char_traits<wchar_t>, std::allocator<wchar_t>>::substr(int, int)
lea rax, [rbp+1020h]
mov rcx, rax
call _Z15SearchFilesMS7_cxx112basic_stringIwSt11char_traitsIwESaIwEEE ; SearchFiles(std::__cxx11::basic_string<wchar_t, std::char_traits<wchar_t>, std::allocator<wchar_t>> const&)
lea rax, [rbp+1020h]
mov rcx, rax
call _ZN57_cxx112basic_stringIwSt11char_traitsIwESaIwEEDiv ; std::__cxx11::basic_string<wchar_t, std::char_traits<wchar_t>, std::allocator<wchar_t>>::basic_string()
add dword ptr [rbp+104Ch], 1

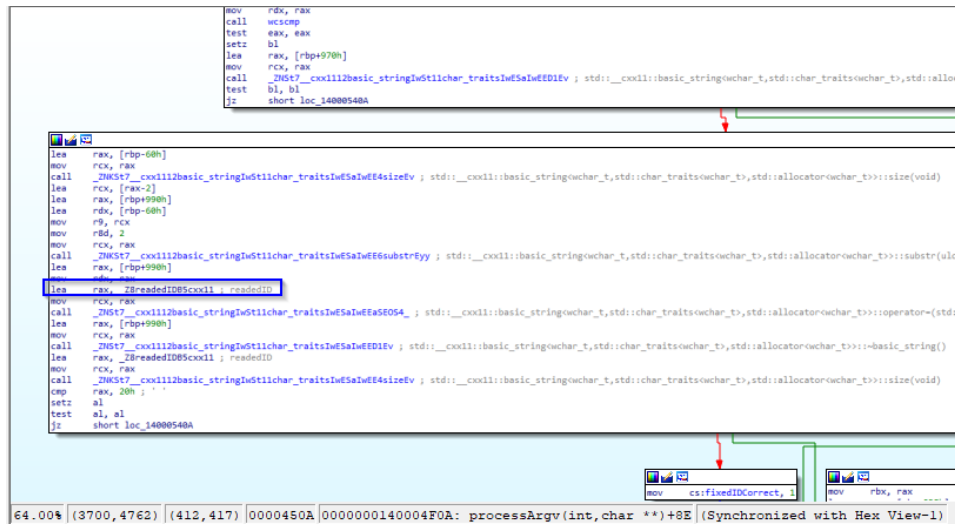
```

64.00% (3579, 1889) (254, 407) 0000401E 0000000140004A1E: searchFilesThreadControl(std: (Synchronized with Hex View-1)

Qui un riferimento all'oggetto contenente la chiave pubblica:

L'operazione di lettura del file C:\ProgramData\ntuser.dat risulta fondamentale per la decriptazione del ransomware stesso:

Qui, difatti, il riferimento all'ID di encryption del sample:



```

mov     rdx, rax
call   wcsncpy
test   eax, eax
setz   bl
lea    rax, [rbp+970h]
mov   rdx, rax
call   _ZN6t7_cxx112basic_stringIwSt11char_traitsIwESaIwE6substrEyy : std::__cxx11::basic_string_owchar_t, std::char_traits_owchar_t, std::allocator_owchar_t>>::substr(along
test   bl, bl
jz     short loc_14000540A

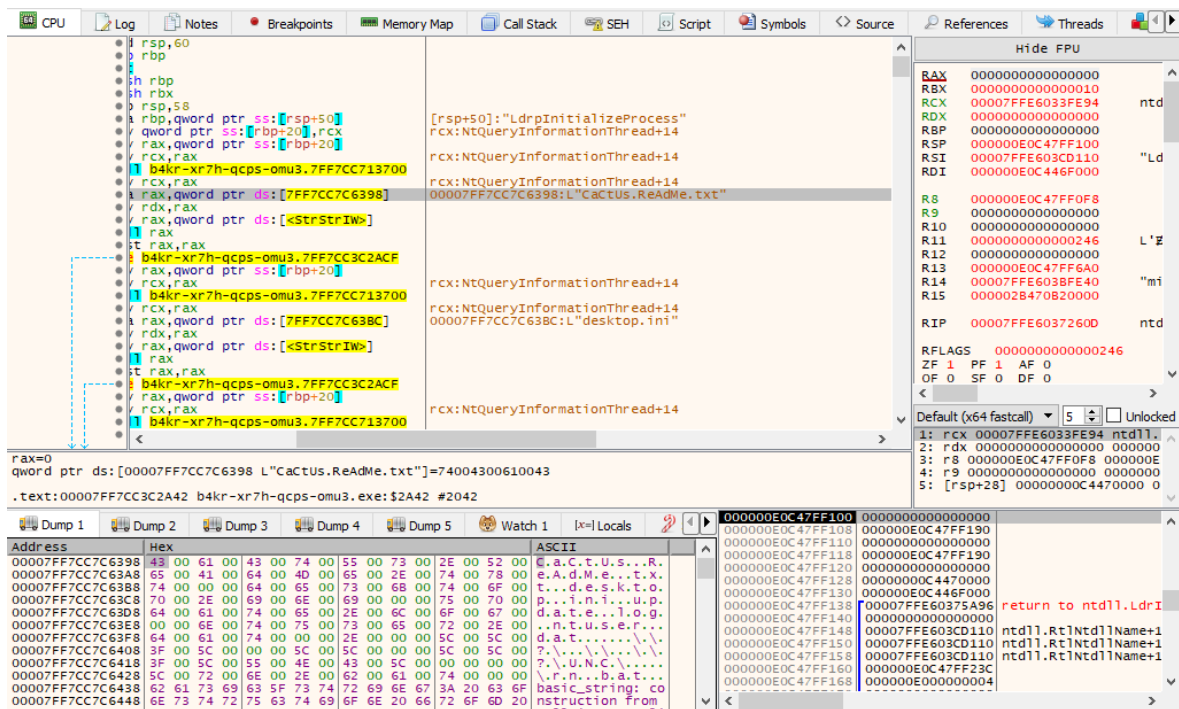
lea    rax, [rbp-60h]
mov   rdx, rax
call   _ZN6t7_cxx112basic_stringIwSt11char_traitsIwESaIwE6sizeEiv : std::__cxx11::basic_string_owchar_t, std::char_traits_owchar_t, std::allocator_owchar_t>>::size(void)
lea    rax, [rax-2]
lea    rdx, [rbp+990h]
mov   r9, rdx
mov   r8d, 2
mov   rax, rax
call   _ZN6t7_cxx112basic_stringIwSt11char_traitsIwESaIwE6substrEyy : std::__cxx11::basic_string_owchar_t, std::char_traits_owchar_t, std::allocator_owchar_t>>::substr(along
lea    rax, [rbp+990h]
lea    rdx, _Z8rededID85cx11 : rededID
mov   rax, rax
call   _ZN6t7_cxx112basic_stringIwSt11char_traitsIwESaIwE6sizeEiv : std::__cxx11::basic_string_owchar_t, std::char_traits_owchar_t, std::allocator_owchar_t>>::size(void)
lea    rax, [rbp+990h]
mov   rax, rax
call   _ZN6t7_cxx112basic_stringIwSt11char_traitsIwESaIwE6substrEyy : std::__cxx11::basic_string_owchar_t, std::char_traits_owchar_t, std::allocator_owchar_t>>::substr(along
lea    rax, [rbp+990h]
mov   rax, rax
call   _ZN6t7_cxx112basic_stringIwSt11char_traitsIwESaIwE6sizeEiv : std::__cxx11::basic_string_owchar_t, std::char_traits_owchar_t, std::allocator_owchar_t>>::size(void)
cmp   rax, 20h
setz   al
test  al, al
jz     short loc_14000540A
  
```

Qui i dettagli del codice esadecimale del sample analizzato, ove si evincono riferimenti a chars e strings management:

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Ascii	
00598DA0	23	00	00	00	4C	5F	4D	5F	63	6F	6E	73	74	72	75	63	#...L_M_construc	
00598DB0	74	3C	63	68	61	72	20	63	6F	6E	73	74	2A	3E	00	0D	t<char const*>..	
00598DC0	D9	07	5F	5A	4E	53	74	37	5F	5F	63	78	78	31	31	31	00_ZNSt7_cxx11	
00598DD0	32	62	61	73	69	63	5F	73	74	72	69	6E	67	49	63	53	2basic_stringIcS	
00598DE0	74	31	31	63	68	61	72	5F	74	72	61	69	74	73	49	63	t1lchar_traitsIc	
00598DF0	45	53	61	49	63	45	45	31	32	5F	4D	5F	63	6F	6E	73	ESaIcEEI2_M_cons	
00598E00	74	72	75	63	74	49	50	4B	63	45	45	76	54	5F	53	38	tructIPKcEEvT_S8	
00598E10	5F	53	74	32	30	66	6F	72	77	61	72	64	5F	69	74	65	_St20forward_ite	
00598E20	72	61	74	6F	72	5F	74	61	67	00	C3	9D	00	00	D8	9D	rator_tag.A..0	
00598E30	00	00	05	E1	1C	00	00	2F	F0	01	00	02	C5	36	02	00	..0á..0/20..0A6.	
00598E40	01	2F	F0	01	00	01	2F	F0	01	00	01	47	23	00	00	00	0/20..0/20..0G#...	
00598E50	13	A8	20	00	00	05	7A	02	07	5F	5A	4E	53	74	37	5F	0'...0z 0_ZNSt7_	
00598E60	5F	63	78	78	31	31	31	32	62	61	73	69	63	5F	73	74	_cxx112basic_st	
00598E70	72	69	6E	67	49	63	53	74	31	31	63	68	61	72	5F	74	ringIcSt1lchar_t	
00598E80	72	61	69	74	73	49	63	45	53	61	49	63	45	45	43	34	raitsIcESaIcEEC4	
00598E90	49	53	33	5F	45	45	50	4B	63	52	4B	53	33	5F	00	01	IS3_EEPKcRKS3..0	
00598EA0	30	9E	00	00	40	9E	00	00	02	C5	36	02	00	01	2F	F0	0!..0!..0A6..0/8	
00598EB0	01	00	01	5C	2D	02	00	00	13	A8	20	00	00	00	05	91	02	0..0~..0'...'..0
00598EC0	07	5F	5A	4E	53	74	37	5F	5F	63	78	78	31	31	31	32	0_ZNSt7_cxx112	
00598ED0	62	61	73	69	63	5F	73	74	72	69	6E	67	49	63	53	74	basic_stringIcSt	
00598EE0	31	31	63	68	61	72	5F	74	72	61	69	74	73	49	63	45	1lchar_traitsIcE	
00598EF0	53	61	49	63	45	45	43	34	49	53	33	5F	45	45	79	63	SaIcEEC4IS3_EEyc	
00598F00	52	4B	53	33	5F	00	01	97	9E	00	00	AC	9E	00	00	02	RKS3..0!..0!..	
00598F10	C5	36	02	00	01	37	56	00	00	01	93	00	00	00	01	5C	A6..07V..0!..0\	
00598F20	2D	02	00	00	05	8D	21	00	00	93	00	00	00	00	58	11	24	-..0!..0!..0X0\$
00598F30	00	00	B4	10	00	00	58	49	23	00	00	E0	1E	00	00	00	..0'..0XI#..0à..	
00598F40	07	55	54	00	00	41	62	61	73	69	63	5F	73	74	72	69	0UT..0Abasic_stri	
00598F50	6E	67	3C	77	63	68	61	72	5F	74	2C	20	73	74	64	3A	ng<wchar_t, std:	
00598F60	3A	63	68	61	72	5F	74	72	61	69	74	73	3C	77	63	68	:char_traits<wch	
00598F70	61	72	5F	74	3E	2C	20	73	74	64	3A	3A	61	6C	6C	6F	ar_t>..std::allo	
00598F80	63	61	74	6F	72	3C	77	63	68	61	72	5F	74	3E	20	3E	cator<wchar_t>>	
00598F90	00	20	05	55	0B	3A	EE	00	00	4E	F9	20	00	00	08	05	..00i..0Nü..00	
00598FA0	C0	0E	7A	A0	00	00	43	67	22	00	00	25	F9	20	00	00	A0z..0Cg'..0Xü..	
00598FB0	05	C7	02	5F	5A	4E	53	74	37	5F	5F	63	78	78	31	31	0C_ZNSt7_cxx11	
00598FC0	31	32	62	61	73	69	63	5F	73	74	72	69	6E	67	49	77	12basic_stringIw	
00598FD0	53	74	31	31	63	68	61	72	5F	74	72	61	69	74	73	49	St1lchar_traitsI	
00598FE0	77	45	53	61	49	77	45	45	31	32	5F	41	6C	6C	6F	63	wESaIwEEI2_Alloc	
00598FF0	5F	68	69	64	65	72	43	34	45	50	77	52	4B	53	33	5F	_hiderC4EPwRKS3..	
00599000	00	91	9F	00	00	A1	9F	00	00	02	1A	37	02	00	01	7A	..0!..0!..07_0z	
00599010	A0	00	00	01	8E	2D	02	00	00	25	F9	20	00	00	05	CB	..0!..0!..0Xü..0E	
00599020	02	5F	5A	4E	53	74	37	5F	5F	63	78	78	31	31	31	32	_ZNSt7_cxx112	
00599030	62	61	73	69	63	5F	73	74	72	69	6E	67	49	77	53	74	basic_stringIwSt	
00599040	31	31	63	68	61	72	5F	74	72	61	69	74	73	49	77	45	1lchar_traitsIwE	

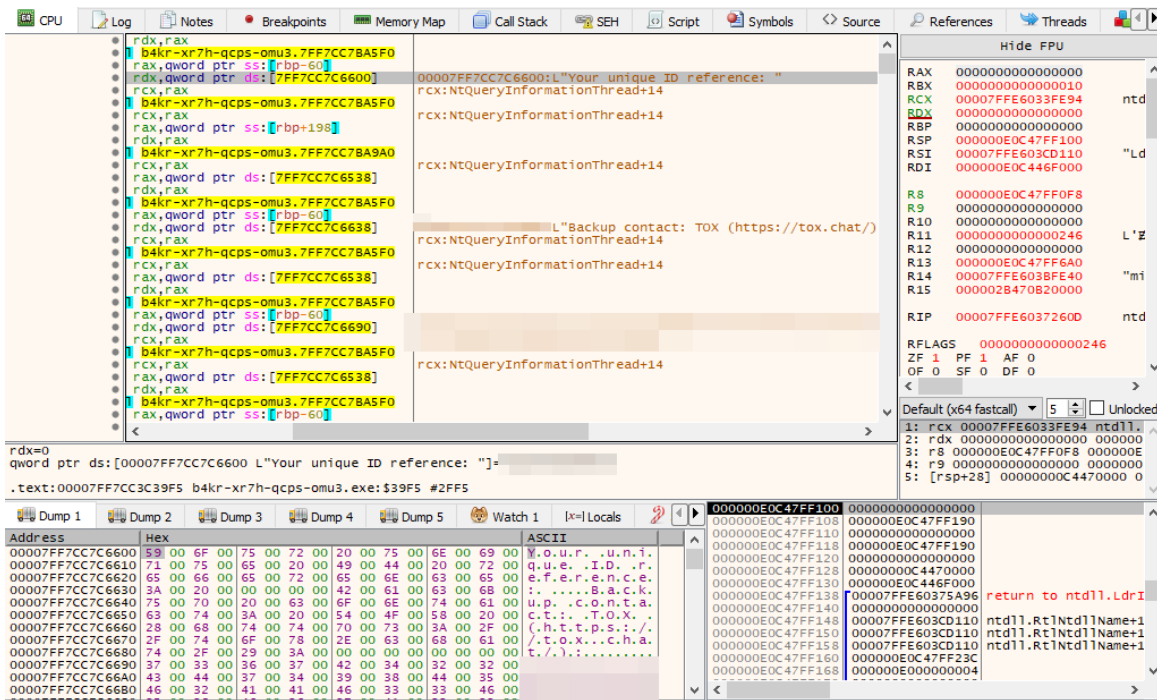
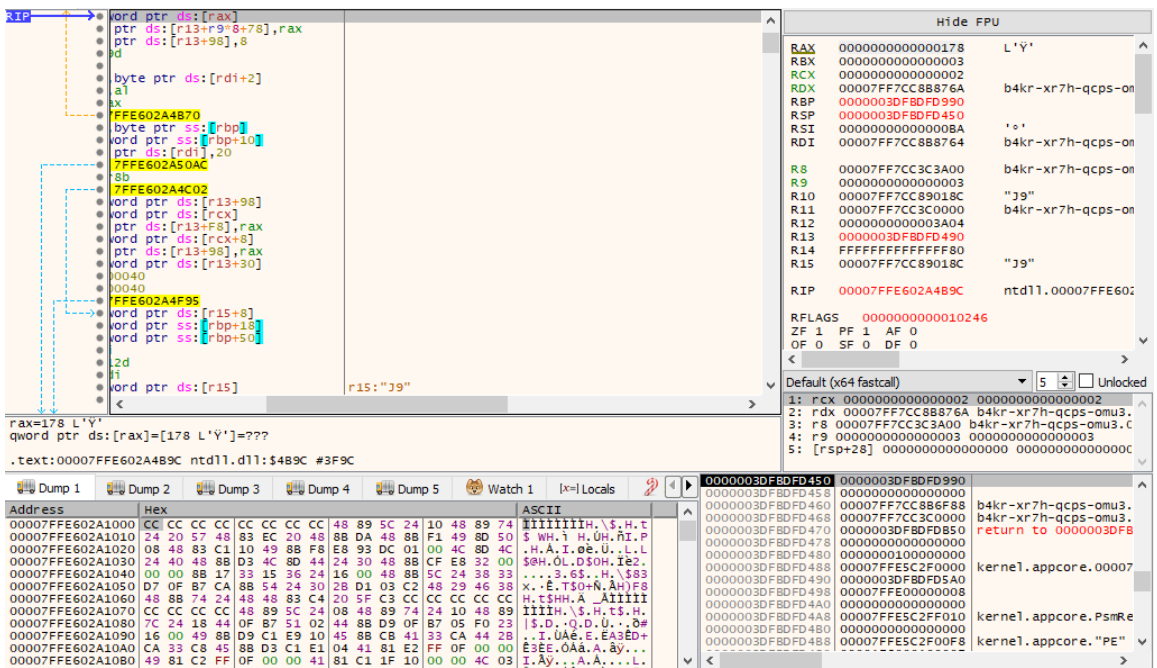
Sessione di debugging

A seguire alcuni dettagli della sessione di debugging effettuata, ove si notano riferimenti al file TXT di readme di Cactus Ransomeware, l'ID univoco dell'infezione ransomware, un riferimento a **TOX chat** per il contatto con i malcoders:

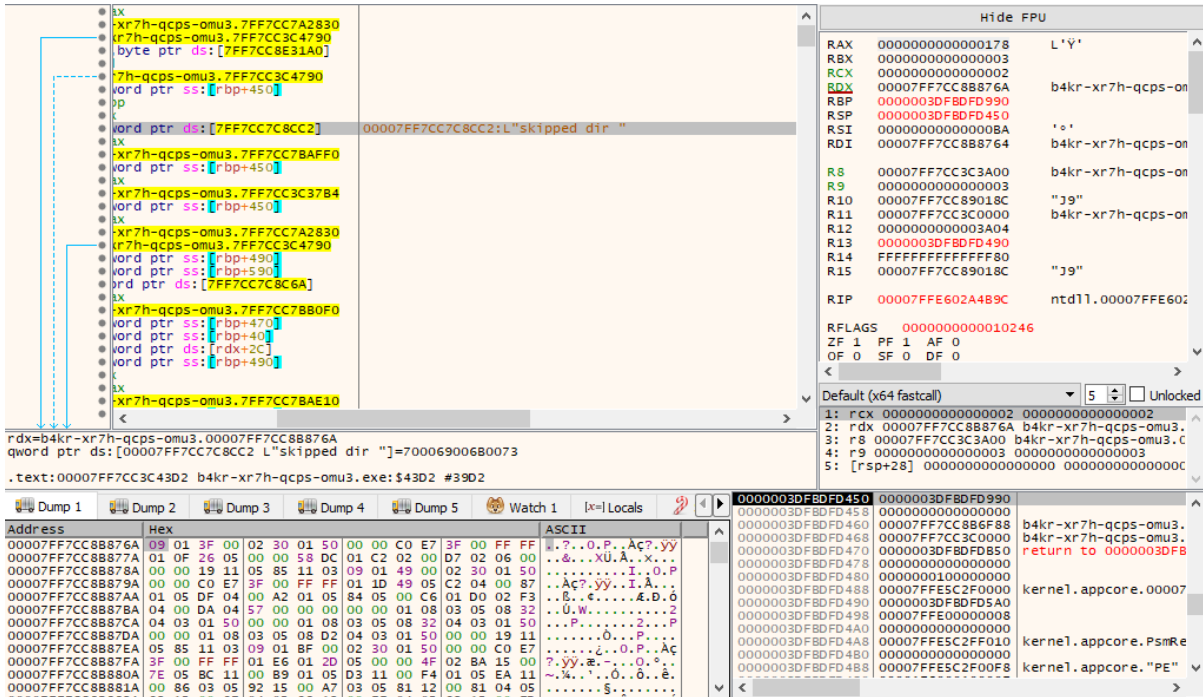


The screenshot shows a debugger interface with several panes:

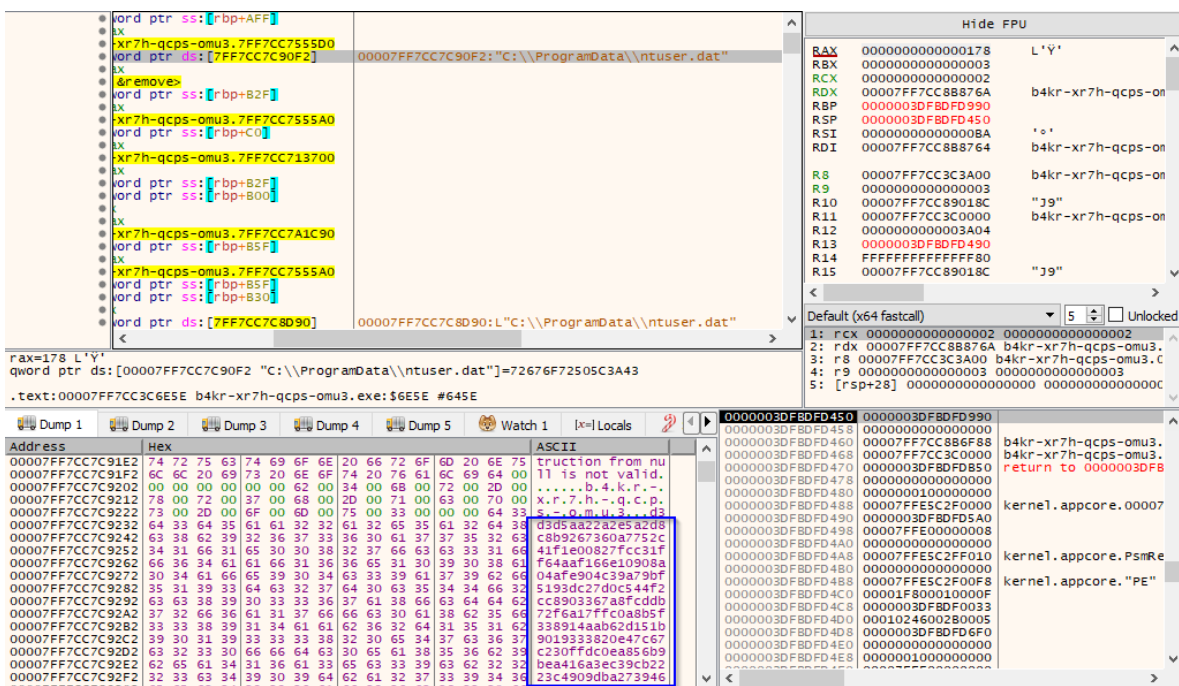
- Registers:** RAX, RBX, RCX, RDX, RBP, RSP, RSI, RDI, R8, R9, R10, R11, R12, R13, R14, R15, RFLAGS.
- Code Window:** Assembly code for `[rsp+50]: "LdrpInitializeProcess"` and `rcx:NtQueryInformationThread+14`. It shows instructions like `mov rax, qword ptr ds:[7FF7CC7C6398]` and `mov rax, qword ptr ds:[*StrStrIW>]`.
- Memory Dump:** A table showing memory addresses, hex values, and ASCII characters. The ASCII column contains the string `e.a.c.t.u.s..r` and `e.a.d.m.e..t.x`.
- Watch Window:** Shows the value of `rax=0` and `qword ptr ds:[00007FF7CC7C6398 L"Cactus.ReAdMe.txt"]=74004300610043`.

Nel caso in cui vengano presi in considerazione files e folders da non cifrare essi vengono "saltati" dal processo di encryption.



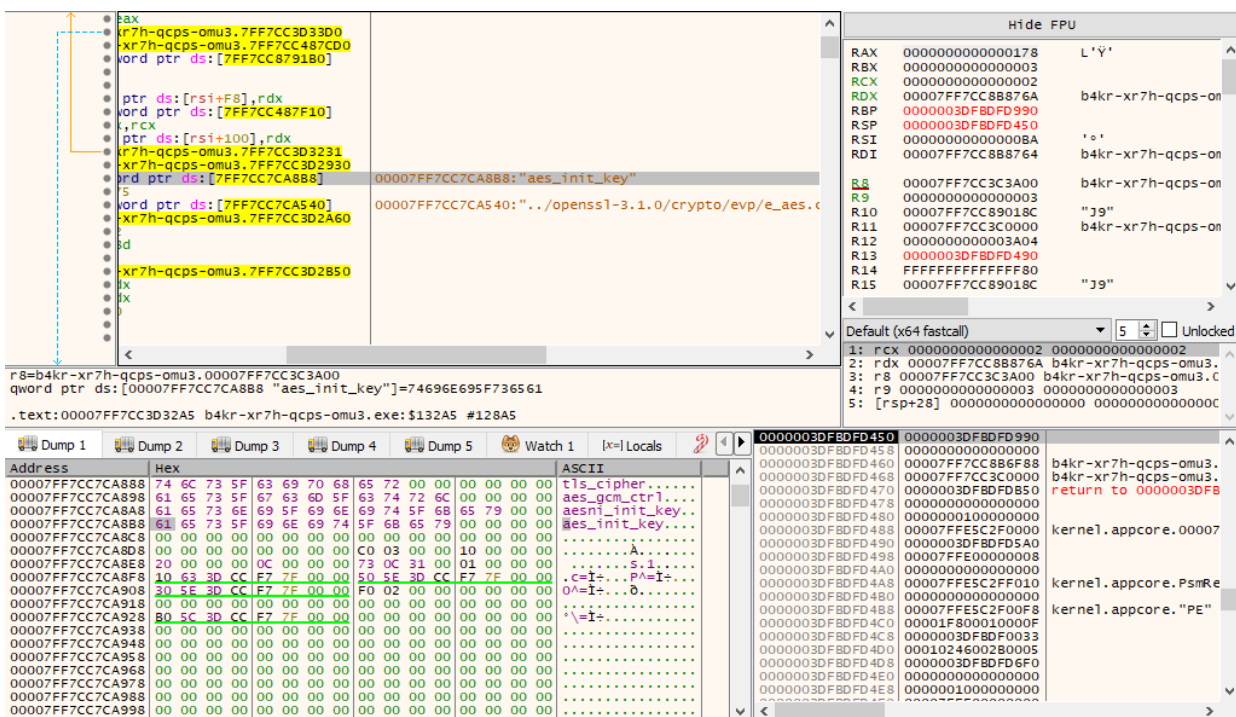
Il file C:\ProgramData\ntuser.dat contiene una stringa randomica che fa riferimento alla stringa di encryption del ransomware sample:



Address	Hex	ASCII
00007FF7CC7C9872	39 31 66 64 36 31 66 36 35 61 65 35 64 66 61 39	91fd61f65ae5dfa9
00007FF7CC7C9882	61 33 35 35 30 66 66 66 35 31 00 4F 4C 69 33 62	a3550fff51.0Li3b
00007FF7CC7C9892	54 4E 36 65 68 5A 43 59 37 6A 64 00 00 63 00	TNgekZCY7jd...c.
00007FF7CC7C98A2	41 00 63 00 54 00 75 00 53 00 2E 00 72 00 65 00	A.c.T.u.S...r.e.
00007FF7CC7C98B2	61 00 64 00 6D 00 65 00 2E 00 74 00 78 00 74 00	a.d.m.e...t.x.t.
00007FF7CC7C98C2	00 00 00 00 00 00 00 00 00 00 00 00 2E 2E	
00007FF7CC7C98D2	2F 6F 70 65 6E 73 73 6C 2D 33 2E 31 2E 30 2F 63	/openssl-3.1.0/c
00007FF7CC7C98E2	72 79 70 74 6F 2F 62 69 6F 2F 62 69 6F 5F 6C 69	rypto/bio/bio_li
00007FF7CC7C98F2	62 2E 63 00 20 00 00 00 00 00 00 00 42 49	b.c.BI
00007FF7CC7C9902	4F 5F 64 6F 5F 63 6F 6E 6E 65 63 74 5F 72 65 74	O.do_connect_ret
00007FF7CC7C9912	72 79 00 00 00 00 42 49 4F 5F 77 61 69 74 00 00	ry...BIO_wait..
00007FF7CC7C9922	00 00 00 00 00 00 42 49 4F 5F 66 69 6E 64 5F 74BIO_find_t
00007FF7CC7C9932	79 70 65 00 00 00 00 00 00 00 00 00 42 49	ype.....BI
00007FF7CC7C9942	4F 5F 63 61 6C 6C 62 61 63 68 5F 63 74 72 6C 00	O_callback_ctrl.
00007FF7CC7C9952	00 00 00 00 00 00 42 49 4F 5F 63 74 72 6C 00BIO_ctrl..
00007FF7CC7C9962	00 00 00 00 00 00 42 49 4F 5F 67 65 74 5F 6C 69BIO_get_li
00007FF7CC7C9972	6E 65 00 00 00 00 42 49 4F 5F 67 65 74 73 00 00	ne...BIO_gets..
00007FF7CC7C9982	00 00 00 00 00 00 42 49 4F 5F 70 75 74 73 00 00BIO_puts..

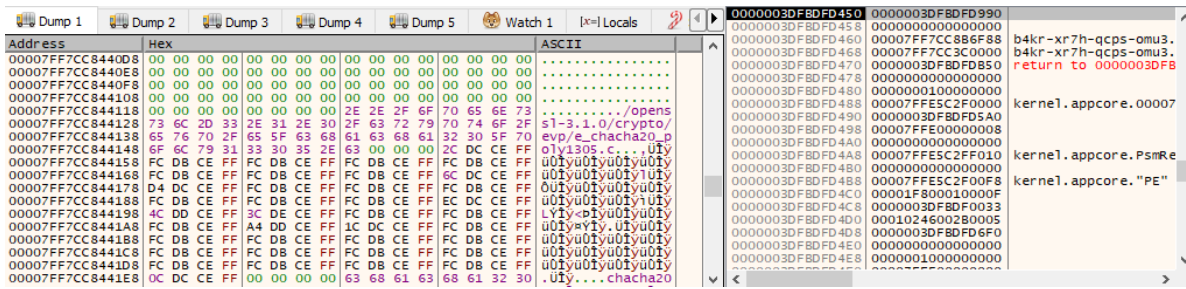

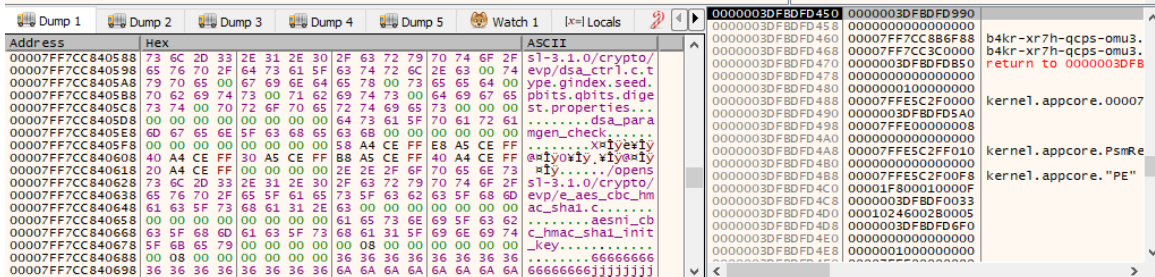
Qui i dettagli ASCII ed esadecimale dei processi presi in considerazione dall'infection kill chain e il richiamo della funzione di AES_init_key:

Address	Hex	ASCII
00007FF7CC7C60F0	65 00 78 00 65 00 00 00 4C 00 6F 00 67 00 6F 00	e.x.e...L.o.g.o.
00007FF7CC7C6100	6E 00 55 00 49 00 2E 00 65 00 78 00 65 00 00 00	n.U.I...e.x.e...
00007FF7CC7C6110	53 00 65 00 61 00 72 00 63 00 68 00 55 00 49 00	S.e.a.r.c.h.U.I.
00007FF7CC7C6120	2E 00 65 00 78 00 65 00 00 6C 00 73 00 61 00	..e.x.e...l.s.a.
00007FF7CC7C6130	73 00 73 00 2E 00 65 00 78 00 65 00 63 00	s.s...e.x.e...c.
00007FF7CC7C6140	73 00 72 00 73 00 73 00 2E 00 65 00 78 00 65 00	s.r.s.s...e.x.e.
00007FF7CC7C6150	00 00 73 00 6D 00 73 00 73 00 2E 00 65 00 78 00	..s.m.s.s...e.x.
00007FF7CC7C6160	65 00 00 00 77 00 69 00 6E 00 6C 00 6F 00 67 00	e...w.i.n.l.o.g.
00007FF7CC7C6170	6F 00 6E 00 2E 00 65 00 78 00 65 00 00 73 00	o.n...e.x.e...s.
00007FF7CC7C6180	65 00 72 00 76 00 69 00 63 00 65 00 73 00 2E 00	e.r.v.i.c.e.s...s.
00007FF7CC7C6190	65 00 78 00 65 00 00 00 63 00 6F 00 6E 00 68 00	e.x.e...c.o.n.h.
00007FF7CC7C61A0	6F 00 73 00 74 00 2E 00 65 00 78 00 65 00 00 00	o.s.t...e.x.e...
00007FF7CC7C61B0	24 00 72 00 65 00 63 00 79 00 63 00 6C 00 65 00	\$.r.e.c.y.c.l.e.
00007FF7CC7C61C0	2E 00 62 00 69 00 6E 00 00 00 00 00 00 00 00 00	..b.i.n.....
00007FF7CC7C61D0	73 00 79 00 73 00 74 00 65 00 6D 00 20 00 76 00	s.y.s.t.e.m...v.
00007FF7CC7C61E0	6F 00 6C 00 75 00 6D 00 65 00 20 00 69 00 6E 00	o.l.u.m.e...i.n.
00007FF7CC7C61F0	66 00 6F 00 72 00 6D 00 61 00 74 00 69 00 6F 00	f.o.r.m.a.t.i.o.
00007FF7CC7C6200	6E 00 00 00 77 00 69 00 6E 00 64 00 6F 00 77 00	n...w.i.n.d.o.w.



Qui possiamo notare encryption attributes di *OpenSSL* e *ChaCha20Poly1305*:

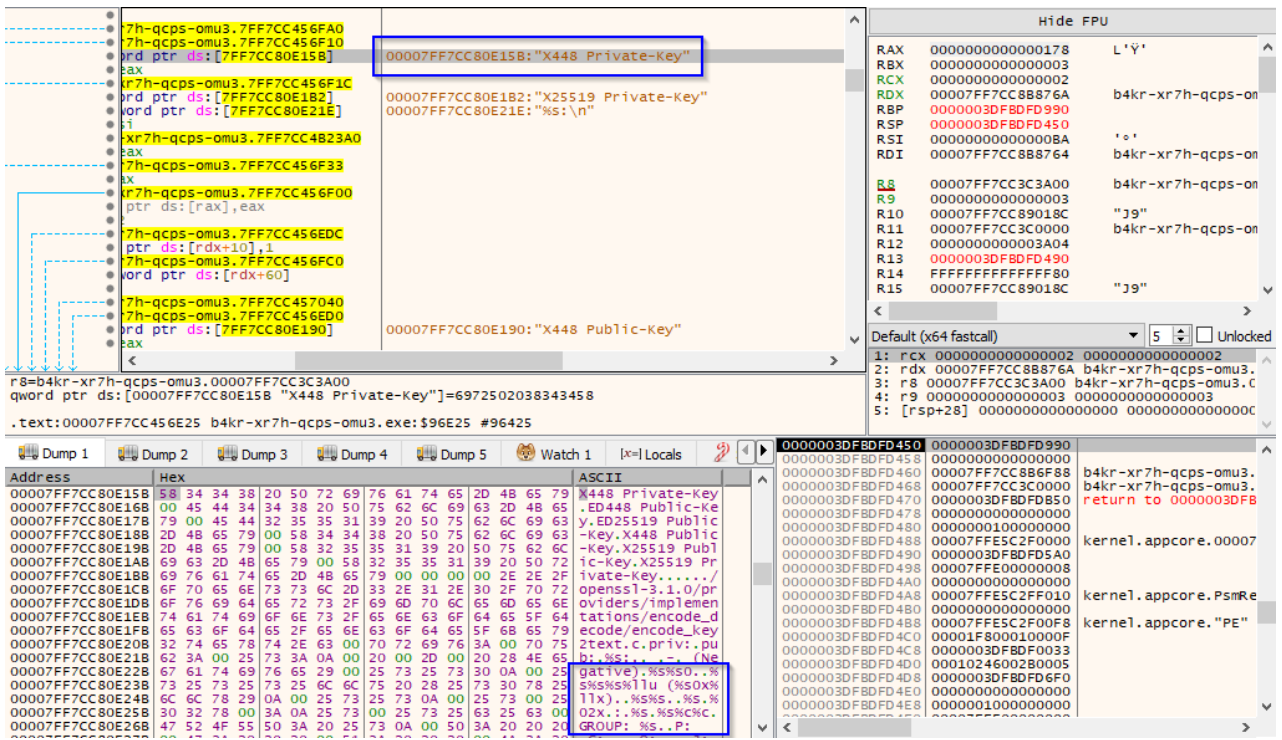
Address	Hex	ASCII
00007FF7CC856698	D9 63 85 CC E7 7F 00 00 7A 00 80 05 00 00 00 00	Ù.Ï+...z.....
00007FF7CC8566A8	EB 63 85 CC E7 7F 00 00 00 00 00 00 00 00 00 00	ë.Ï+.....
00007FF7CC8566B8	00 00 00 00 00 00 00 00 2E 2E 2F 6F 70 65 6E 73/opens
00007FF7CC8566C8	73 6C 2D 33 2E 31 2E 30 2F 63 72 79 70 74 6F 2F	s1-3.1.0/crypto/
00007FF7CC8566D8	78 35 30 39 2F 78 35 30 39 5F 6C 75 2E 63 00 00	x509/x509_lu.c..
00007FF7CC8566E8	00 00 00 00 00 00 00 00 58 35 30 39 5F 53 54 4FX509_STO
00007FF7CC8566F8	52 45 5F 67 65 74 31 5F 61 6C 6C 5F 63 65 72 74	RE_get1_all_cert
00007FF7CC856708	73 00 00 00 00 00 00 00 58 35 30 39 5F 4F 42 4A	s.....X509_OB
00007FF7CC856718	45 43 54 5F 6E 65 77 00 58 35 30 39 5F 53 54 4F	ECT_new.X509_STO
00007FF7CC856728	52 45 5F 61 64 64 5F 63 72 6C 00 00 00 00 00	RE_add_cr1.....
00007FF7CC856738	00 00 00 00 00 00 00 00 58 35 30 39 5F 53 54 4FX509_STO
00007FF7CC856748	52 45 5F 61 64 64 5F 63 65 72 74 00 00 00 00	RE_add_cert.....
00007FF7CC856758	00 00 00 00 00 00 00 00 58 35 30 39 5F 53 54 4FX509_STO
00007FF7CC856768	52 45 5F 61 64 64 5F 6C 6F 6F 68 75 70 00 00 00	RE_add_lookup...
00007FF7CC856778	58 35 30 39 5F 53 54 4F 52 45 5F 6E 65 77 00 00	X509_STORE_new..
00007FF7CC856788	00 00 00 00 00 00 00 00 58 35 30 39 5F 4C 4F 4FX509_LOO
00007FF7CC856798	48 55 50 5F 6E 65 77 00 2E 2E 2F 6F 70 65 6E 73	KUP_new.../opens
00007FF7CC8567A8	73 6C 2D 33 2E 31 2E 30 2F 63 72 79 70 74 6F 2F	s1-3.1.0/crypto/

Si notino i dettagli in merito all'encryption routine Blake2 e l'inserimento della chiave privata con i tipi di dati **%s (string)** e **%C (caratteri)**.

Address	Hex	ASCII
00007FF7CC817FE8	FF FF FF FF FF FF FF FF 00 00 00 00 00 00 00 00	yyyyyyyy.....
00007FF7CC817FF8	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00007FF7CC818008	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00007FF7CC818018	00 00 00 00 00 00 00 00 73 69 7A 65 00 62 6C 6Fsize.blo
00007FF7CC818028	63 68 2D 73 69 7A 65 00 2E 2E 2F 6F 70 65 6E 73	ck-size.../opens
00007FF7CC818038	73 6C 2D 33 2E 31 2E 30 2F 70 72 6F 76 69 64 65	s1-3.1.0/provide
00007FF7CC818048	72 73 2F 69 6D 70 6C 65 6D 65 6E 74 61 74 69 6F	rs/implematio
00007FF7CC818058	6E 73 2F 6D 61 63 73 2F 62 6C 61 68 65 32 5F 6D	ns/macs/blake2_m
00007FF7CC818068	61 63 5F 69 6D 70 6C 2E 63 00 68 65 79 00 63 75	ac_impl.c.key.cu
00007FF7CC818078	73 74 6F 6D 00 73 61 6C 74 00 00 00 00 00 00 00	stom.salt.....
00007FF7CC818088	00 00 00 00 00 00 00 00 62 6C 61 68 65 32 5F 6Dblake2_m
00007FF7CC818098	61 63 5F 69 6E 69 74 00 62 6C 61 68 65 32 5F 73	ac_init.blake2_s
00007FF7CC8180A8	65 74 68 65 79 00 00 00 62 6C 61 68 65 32 5F 6D	etkey...blake2_m
00007FF7CC8180B8	61 63 5F 73 65 74 5F 63 74 78 5F 70 61 72 61 6D	ac_set_ctx_param
00007FF7CC8180C8	73 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00	s.....
00007FF7CC8180D8	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00007FF7CC8180E8	E0 D3 46 CC F7 7F 00 00 02 00 00 00 00 00 00 00	aöFi+.....
00007FF7CC8180F8	10 D3 46 CC F7 7F 00 00 03 00 00 00 00 00 00 00	.öFi+.....

Address	Hex	ASCII
00007FF7CC809520	64 68 5F 74 6F 5F 53 75 62 6A 65 63 74 50 75 62	dh_to_SubjectPub
00007FF7CC809530	6C 69 63 48 65 79 49 6E 66 6F 5F 70 65 6D 5F 65	licKeyInfo_pem_e
00007FF7CC809540	6E 63 6F 64 65 00 00 00 00 00 00 00 00 00 00 00	ncode.....
00007FF7CC809550	64 68 5F 73 70 68 69 5F 70 75 62 5F 74 6F 5F 64	dh_spki_pub_to_d
00007FF7CC809560	65 72 00 00 00 00 00 00 00 00 00 00 00 00 00 00	er.....
00007FF7CC809570	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00007FF7CC809580	64 68 5F 74 6F 5F 53 75 62 6A 65 63 74 50 75 62	dh_to_SubjectPub
00007FF7CC809590	6C 69 63 48 65 79 49 6E 66 6F 5F 64 65 72 5F 65	licKeyInfo_der_e
00007FF7CC8095A0	6E 63 6F 64 65 00 00 00 00 00 00 00 00 00 00 00	ncode.....
00007FF7CC8095B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00007FF7CC8095C0	64 68 5F 74 6F 5F 50 72 69 76 61 74 65 48 65 79	dh_to_PrivateKey
00007FF7CC8095D0	49 6E 66 6F 5F 70 65 6D 5F 65 6E 63 6F 64 65 00	Info_pem_encode.
00007FF7CC8095E0	64 68 5F 74 6F 5F 50 72 69 76 61 74 65 48 65 79	dh_to_PrivateKey
00007FF7CC8095F0	49 6E 66 6F 5F 64 65 72 5F 65 6E 63 6F 64 65 00	Info_der_encode.
00007FF7CC809600	64 68 5F 74 6F 5F 45 6E 63 72 79 70 74 65 64 50	dh_to_EncryptedP
00007FF7CC809610	72 69 76 61 74 65 48 65 79 49 6E 66 6F 5F 70 65	rivateKeyInfoPe
00007FF7CC809620	6D 5F 65 6E 63 6F 64 65 00 00 00 00 00 00 00 00	m_encode.....
00007FF7CC809630	70 72 65 70 61 72 65 5F 64 68 5F 70 61 72 61 6D	prepare_dh_param

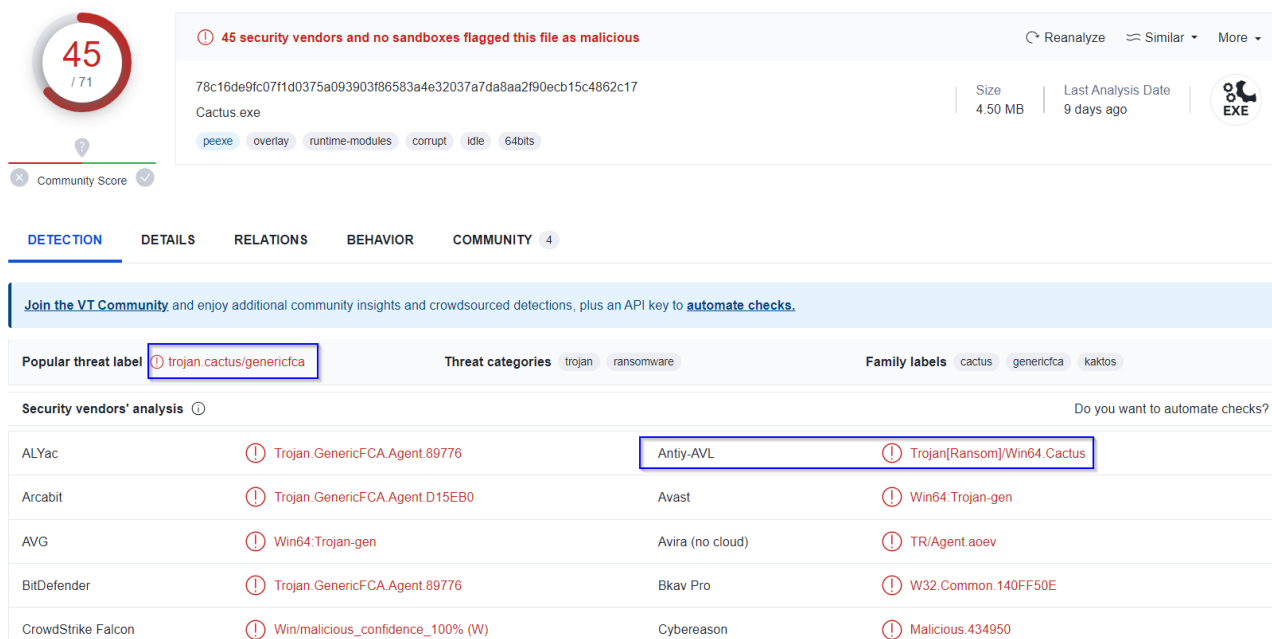


The screenshot displays a debugger interface with three main panes:

- Assembly View:** Shows instructions for loading and storing private keys. Key instructions include:
 - `00007FF7CC80E15B: "X448 Private-Key"`
 - `00007FF7CC80E1B2: "X25519 Private-Key"`
 - `00007FF7CC80E21E: "%s:\n"`
 - `00007FF7CC80E190: "X448 Public-Key"`
- Register View:** Shows registers RAX through R15. RAX contains `0000000000000178`, RBX contains `0000000000000003`, RCX contains `0000000000000002`, and so on.
- Stack View:** Shows memory addresses and their contents. The stack contains several instances of `0000000000000000` and `0000003DFBDF450`.

Threat research

Il sample è riconosciuto come malevolo dalle fonti OSINT ed identificato come **trojan.cactus/genericfca**:



45 / 71 Community Score

45 security vendors and no sandboxes flagged this file as malicious

78c16de9fc07f1d0375a093903f86583a4e32037a7da8aa2f90ecb15c4862c17
Cactus.exe

Size: 4.50 MB | Last Analysis Date: 9 days ago

peexe overlay runtime-modules corrupt idle 64bits

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 4

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label: trojan.cactus/genericfca

Threat categories: trojan ransomware

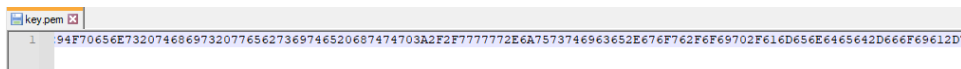
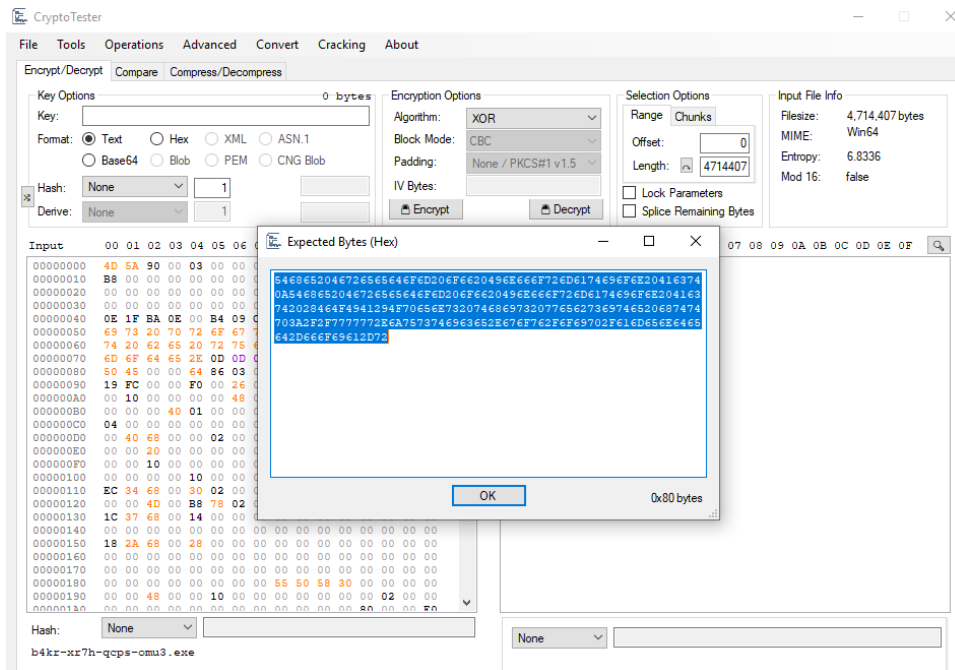
Family labels: cactus genericfca kaktos

Security vendors' analysis

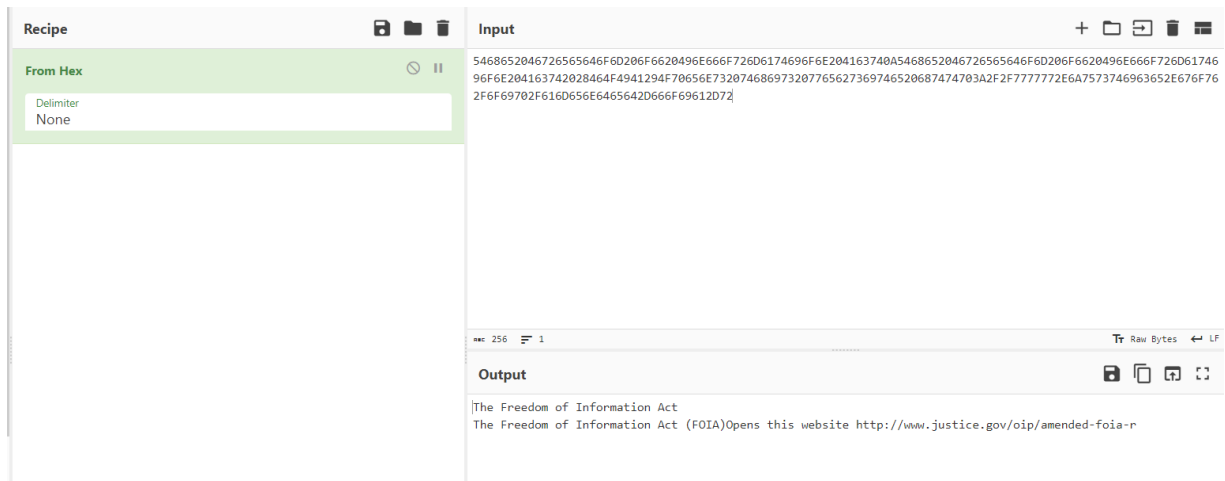
Vendor	Detection	Vendor	Detection
ALYac	Trojan.GenericFCA.Agent.89776	Antiy-AVL	Trojan[Ransom]/Win64.Cactus
Arcabit	Trojan.GenericFCA.Agent.D15EB0	Avast	Win64.Trojan-gen
AVG	Win64:Trojan-gen	Avira (no cloud)	TR/Agent.aoev
BitDefender	Trojan.GenericFCA.Agent.89776	Bkav Pro	W32.Common.140FF50E
CrowdStrike Falcon	Win/malicious_confidence_100% (W)	Cybereason	Malicious.434950

Qui un dettaglio di un key finding attempt mediante la stringa randomica inserita nel file **C:\ProgramData\ntuser.dat**:

Offset	Size	Type	String
306	00405730	00000100 U	5468652046726565646F6D206F66620496E666F726D6174696F6E204163740A5468652046726...



Effettuando una decodifica dall'esadecimale della stringa in questione è possibile identificare il dominio di Freedom of Information Act ed il dominio justice[.]gov. Tuttavia, con i pattern successivi essa dovrebbe contenere la sottstringa utilizzata come chiave pubblica per la cifratura del ransomware.



Cactus Ransomware utilizza tecniche di Anti-VM basate sul CPUID:

Environment Awareness

The input sample contains a known anti-VM trick

details Found VM detection artifact "CPUID trick" in "sample.bin" (Offset: 105697)

source Binary File

relevance 5/10

ATT&CK ID T1497 ([Show technique in the MITRE ATT&CK™ matrix](#))

A seguire un dettaglio di caricamento dinamico del modulo di encryption bcryptprimitives.dll in un contesto di esecuzione:

Loads the cryptographic module DLL

details "<Input Sample>.exe" loaded module "%WINDIR%\System32\bcryptprimitives.dll" at 41380000

source Loaded Module

ATT&CK ID T1027 ([Show technique in the MITRE ATT&CK™ matrix](#))

Qui un dettaglio di enumerazione di librerie DLL utilizzate, nello specifico RSTRMGR.DLL (per gestire eventuali files già in uso ad altri processi), NCRYPT.DLL e BCRYPT.DLL, WSOCK32.DLL (che gestisce connessioni sockets, con tutta probabilità adoperata anche in fase di connessioni

Command and Control), FLTLIB.DLL (utilizzabile per filtri e ricerche, anche in sede di files enumeration).

Tries to access non-existent files (executable)

details "<Input Sample>.exe" trying to access non-existent file "C:\RSTRMGR.DLL"
"<Input Sample>.exe" trying to access non-existent file "C:\NCRYPT.DLL"
"<Input Sample>.exe" trying to access non-existent file "C:\BCRYPT.DLL"
"<Input Sample>.exe" trying to access non-existent file "C:\NTASN1.DLL"
"<Input Sample>.exe" trying to access non-existent file "C:\WSOCK32.DLL"
"<Input Sample>.exe" trying to access non-existent file "C:\FLTLIB.DLL"
"<Input Sample>.exe" trying to access non-existent file "C:\CRYPTBASE.DLL"

source API Call

relevance 3/10

ATT&CK ID T1083 ([Show technique in the MITRE ATT&CK™ matrix](#))

Qui le funzionalità di sockets connections:

Network Related

Contains ability to communicate with network (APIs)

details Found api string "accept" (Indicator: "accept"; Source: "00000000-00004400-00000C1F-78593735")
Found api string "connect" (Indicator: "connect"; Source: "00000000-00004400-00000C1F-78594094")
Found api string "listen" (Indicator: "listen"; Source: "00000000-00004400-00000C1F-78594795")
Found api string "socket" (Indicator: "socket"; Source: "00000000-00004400-00000C1F-78595669")

source File/Memory

relevance 1/10

ATT&CK ID T1071 ([Show technique in the MITRE ATT&CK™ matrix](#))

Qui un'evidenza della lettura delle informazioni relative al language pack installato sulla macchina con il fine di identificare la potenziale nazionalità delle vittime:

Unusual Characteristics

Reads the windows installation language

details "<Input Sample>.exe" (Path: "HKLM\SYSTEM\CONTROLSET001\CONTROL\NLS\LANGUAGE GROUPS"; Key: "1")

source Registry Access

relevance 2/10

ATT&CK ID T1614.001 ([Show technique in the MITRE ATT&CK™ matrix](#))

A seguire i dettagli di un Pulse (il quale rappresenta un IOC container) riferibile a diversi indicatori associati a Cactus Ransomware:



Cactus Ransomware IOCs | PasteNet - Powerful Notetaking

CREATED 2 MONTHS AGO by Bheeshmar | Public | TLP: Green

Cactus Ransomware is a Ransomware which is active since March 2023. They have launched their DLS on DarkWeb in July 2023.

REFERENCES: <http://pastenet.com/oDU2sokV/>
<https://twitter.com/RakeshKrish12/status/1682236940073185282>

TAGS: pastenet, pastebin, paste, manufacturing, iocs, cactus, ransomware, sonar, malware, databreach, dataleak

INDUSTRIES: Transportation, Food, Manufacturing, Construction, Marketing, Hardware, Nature

TARGETED COUNTRIES: United States of America, Italy, United Kingdom of Great Britain and Northern Ireland, Portugal, France, India, Switzerland

MALWARE FAMILIES: CACTUS, Cactus

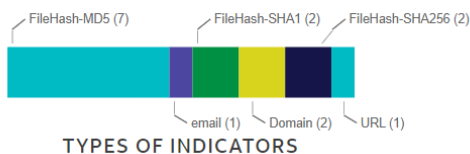
ENDPOINT SECURITY Scan your endpoints for IOCs from this Pulse!

Indicators of Compromise (15)

Related Pulses (1)

Comments (0)

History (0)



E' possibile avere contezza di un ulteriore dominio TOR, nello specifico sonarmsng5vzwqezlvtu2iiwwdn3dxkhotftikhowpfjuzg7p3ca5eid[.]onion e l'indirizzo e-mail di contatto cactus787835[.]proton[.]me:

TYPE	INDICATOR	ROLE	TITLE	ADDED	ACTIVE	RELATED PULSES
email	cactus787835@proton.me			Jul 21, 2023, 4:00:45 AM		0
domain	sonarmsng5vzwqezlvту2iiwwdn3dxkhotftikhowpfjuzg7p3ca5eid.onion			Jul 21, 2023, 4:00:45 AM		1
domain	cactusbloguodvqjmnzlwetjlpj6aggc6iocwhuupb47laukux7ckid.onion			Jul 21, 2023, 4:00:45 AM		0
URL	http://sonarmsng5vzwqezlvту2iiwwdn3dxkhotftikhowpfjuzg7p3ca5eid.onion/cont...			Jul 21, 2023, 4:00:45 AM		0
FileHash-SHA256	c52ad663ff29e146de6b7b20d834304202de7120e93a93de1de1cb1d56190bfd		RC6_Constants	Jul 21, 2023, 4:00:45 AM		0
FileHash-SHA256	78c16de9fc07f1d0375a093903f86583a4e32037a7da8aa2f90ecb15c4862c17		RC6_Constants	Jul 21, 2023, 4:00:45 AM		0
FileHash-SHA1	cb570234349507a204c558fc8c4ecf713e2c0ac3		RC6_Constants	Jul 21, 2023, 4:00:45 AM		0
FileHash-SHA1	173f9b0db97097675a028b4b877630adc7281d2f		RC6_Constants	Jul 21, 2023, 4:00:45 AM		0
FileHash-MD5	eba1596272ff695a1219b1380468293a			Jul 21, 2023, 4:00:45 AM		0
FileHash-MD5	e28db6a65da2ebcf304873c9a5ed086d		RC6_Constants	Jul 21, 2023, 4:00:45 AM		0

TYPE	INDICATOR	ROLE	TITLE	ADDED	ACTIVE	RELATED PULSES
FileHash-MD5	de6ce47e28337d28b6d29ff61980b2e9		RC6_Constants	Jul 21, 2023, 4:00:45 AM		0
FileHash-MD5	949d9523269604db26065f002feef9ae			Jul 21, 2023, 4:00:45 AM		0
FileHash-MD5	5737cb3a9a6d22e957cf747986eeb1b3			Jul 21, 2023, 4:00:45 AM		0
FileHash-MD5	2611833c12aa97d3b14d2ed541df06b2			Jul 21, 2023, 4:00:45 AM		0
FileHash-MD5	1add9766eb649496bc2fa516902a5965			Jul 21, 2023, 4:00:45 AM		0

IOCs

- 78c16de9fc07f1d0375a093903f86583a4e32037a7da8aa2f90ecb15c4862c17
- cb570234349507a204c558fc8c4ecf713e2c0ac3
- e28db6a65da2ebcf304873c9a5ed086d
- Updates Check Task scheduled task
- CaCtUs.ReAdMe.txt
- cactus[.]mexicomail[.]com
- TOX Chat ID:
7367B422CD7498D5F2AAF33F58F67A332F8520CF0279A5FBB4611E0121AE421AE1D49ACEABB254686
- cactus787835[.]proton[.]me
- cactusbloguodvqjmnzlwetjlpj6aggc6iocwhuupb47laukux7ckid[.]onion
- sonarmsng5vzwqezlvту2iiwwdn3dxkhotftikhowpfjuzg7p3ca5eid[.]onion

Regola YARA

```
rule CactusRule
{
  strings:
    $cactusStr = "CaCtUs.ReAdMe.txt"
    $cactusHex = { 43 61 43 74 55 73 2e 52 65 41 64 4d 65 2e 74 78 74 }

  condition:
    $cactusStr or $cactusHex
}
```

Conclusioni

Cactus Ransomware possiede numerose caratteristiche che, almeno potenzialmente, potrebbero cambiare lo scenario delle nuove infezioni ransomware. I nuovi threats assumerebbero nuove caratteristiche fondamentali di auto-encryption ed il cambio consecutivo di più estensioni dei files criptati, questo per rendere più difficoltosa l'identificazione dei files stessi, sottoposti al processo di cifratura. È inoltre presente una peculiarità relativa all'utilizzo del file C:\ProgramData\ntuser.dat al fine di archiviare la stringa utilizzata come chiave pubblica per la crittografia del malware sample stesso. Il file ntuser.dat rappresenta un elemento utilizzato ad-hoc per la self-encryption del ransomware appositamente con tale filename con lo scopo probabilmente di confondere ed effettuare evasion. Per comprendere la natura ed il funzionamento di Cactus Ransomware possiamo citare due particolarità importanti: esso utilizza il packer UPX, il quale è largamente conosciuto e semplice da "unpackare". Inoltre, i files sottoposti al processo di encryption, vengono divisi in porzioni salvate in microbuffers, probabilmente per velocizzare la gestione dei data streams criptati. Dalle evidenze dei real-world scenarios si evince una volontà da parte di Cactus Ransomware di effettuare due principali tasks di discovery incrociati con lo scopo di riuscire più facilmente ad ottenere informazioni e dettagli in merito alle utenze ed al dominio anche nel caso in cui ci sia la detection (ed eventualmente il blocco preventivo) di una delle metodologie utilizzate, come ad esempio il tool di scansione SoftPerfect Network Scanner.

Riferimenti

[0]: [Cactus Ransomware Group Adds Four New Victims \(privacyaffairs.com\)](#)

[1]: [What is CACTUS Ransomware? \(lepide.com\)](#)