

## **Corso di Codice Malevolo**

Relazione sull'analisi del malware sample2.exe

Candidati:

**Riccardo Astolfi**

**Giacomo Ferro**

**Francesco Gobbi**

# Indice

<b>I</b>	<b>Analisi Statica</b>	<b>2</b>
	Analisi dell'intro	2
	Analisi del manifesto	3
	Analisi della versione	4
	Analisi degli indicatori	5
	Analisi delle sezioni	6
	Analisi delle librerie	7
	Analisi degli import	8
	Analisi delle stringhe	9
	Analisi con PEID	10
	VirusTotal	11
<b>II</b>	<b>Analisi Dinamica</b>	<b>14</b>
	Expiro	14
	Analisi tramite RegShot e ProcMon	16
	Analisi tramite Fakenet e Wireshark	18
	FakeNet	18
	Wireshark	20
<b>III</b>	<b>Reverse Engineering</b>	<b>21</b>
	Offuscamento	21
	Disattivazione Security Center	22
	Salvataggio dei file	23
<b>IV</b>	<b>Conclusioni</b>	<b>25</b>
	Bibliografia	26

## Parte I

# Analisi Statica

*L'analisi statica consiste nel dedurre il comportamento di un software senza eseguirlo, basandosi solo sulla forma, sulla struttura e sul contenuto.*

Il file analizzato è un eseguibile di Windows con estensione `.exe` che riporta l'icona di una calcolatrice e ha nome `sample2.exe`.

Iniziamo l'analisi con **PEStudio** (già presente tra i tool della macchina virtuale) il quale permette l'analisi di file con architettura a 32 bit.

Il programma malevolo non può essere eseguito da riga di comando e presenta un'interfaccia utente della forma di una calcolatrice di Windows.

## Analisi dell'intro

Analizziamo l'*intro* del file in questione che è stato *compilato e debuggato* in data 17 Agosto 2001 alle 21:52:32 e ha una dimensione di 626 KB (*626688 bytes*).

L'**entropia** risulta essere 7.189.

*Nella teoria dell'informazione l'entropia indica il livello di "casualità" ovvero quanto disordinatamente sono disposti i byte di un sistema.*

Il dato ottenuto è tanto più informativo quanto più la probabilità è bassa per la legge di Shannon. In poche parole l'entropia misura il livello di compressione ed/od offuscamento del codice.

Se un file non è compresso/offuscato allora l'entropia sarà bassa e questo significa che l'analisi statica sarà la tecnica più appropriata per l'analisi del codice.

Questa sezione del tool presenta tre tipi di hash:

- md5 : F83C765FB553146712FCF2C6066670B5
- sha1 : A6B849E7A8312F5D7E3D7C96501887F39E3BE512
- sha256 : 34558AC3BFAB17CA1A1FF70860B35296395F1DF7FA8D86B39C56FAECF9C3CFFC

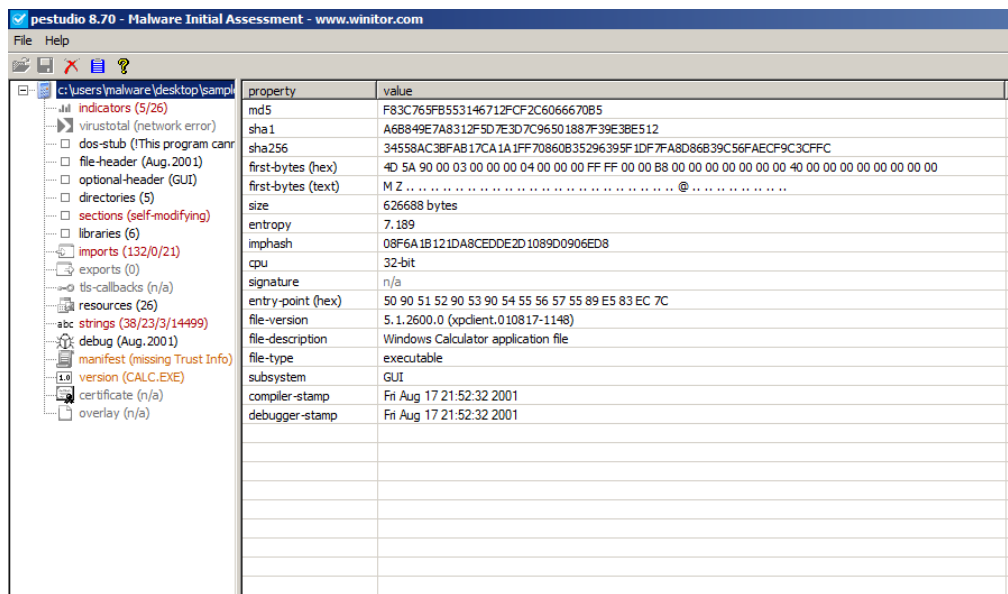


Figura 1: PESTudio - Informazioni del file analizzato

Il malware non è firmato dall'autore. I primi due byte, tradotti in caratteri ASCII, identificano i caratteri "MZ" ovvero la firma caratteristica dei file di tipo PE (Portable Executable).

## Analisi del manifesto

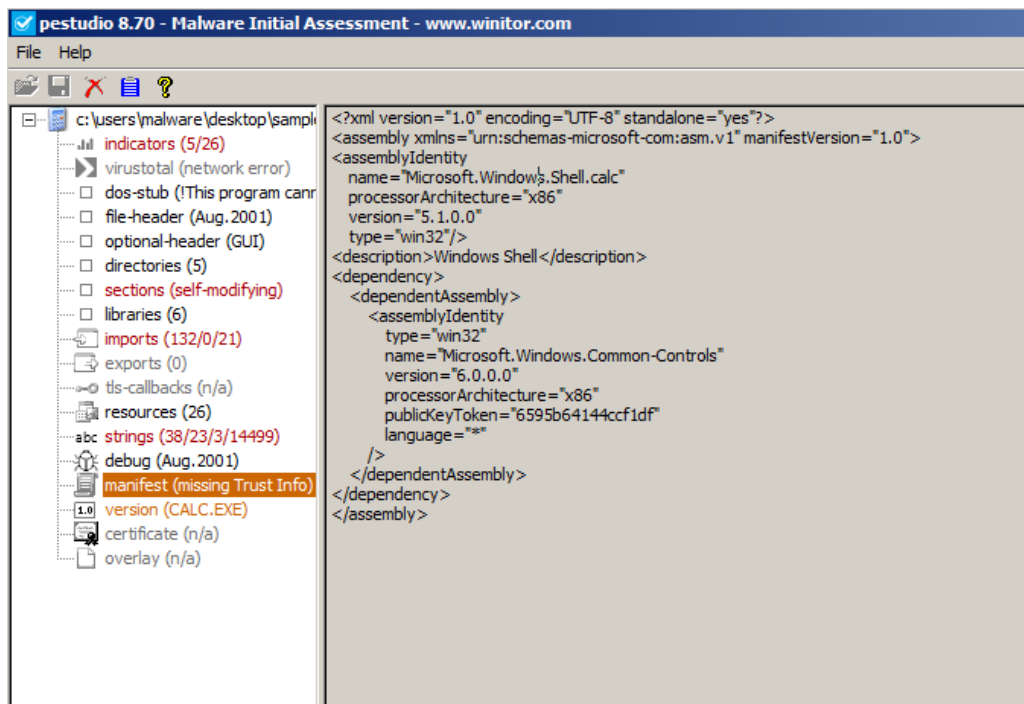


Figura 2: PESTudio - Manifest

Durante l'esecuzione si mostra come una calcolatrice evitando quindi di insospettire l'utente e permettendo l'esecuzione dei comandi in background.

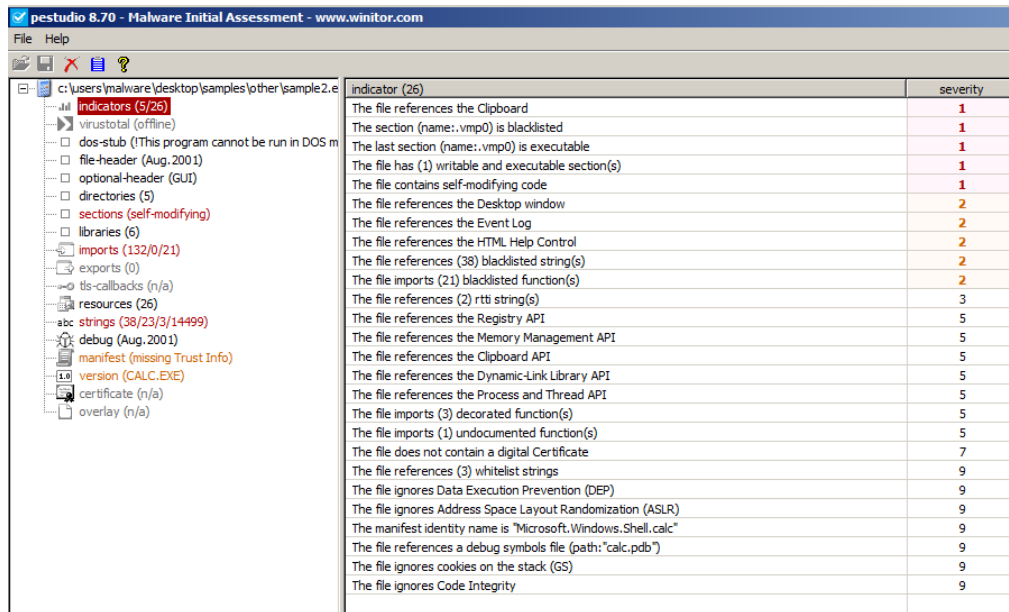
## Analisi della versione



Tra i più importanti ci sono: la codifica dei caratteri unicode (UTF-16), l'ordine dei bit (tipo little endian) e la lingua del programma malevolo (inglese).

4

## Analisi degli indicatori



indicator (26)	severity
The file references the Clipboard	1
The section (name:.vmp0) is blacklisted	1
The last section (name:.vmp0) is executable	1
The file has (1) writable and executable section(s)	1
The file contains self-modifying code	1
The file references the Desktop window	2
The file references the Event Log	2
The file references the HTML Help Control	2
The file references (38) blacklisted string(s)	2
The file imports (21) blacklisted function(s)	2
The file references (2) rtti string(s)	3
The file references the Registry API	5
The file references the Memory Management API	5
The file references the Clipboard API	5
The file references the Dynamic-Link Library API	5
The file references the Process and Thread API	5
The file imports (3) decorated function(s)	5
The file imports (1) undocumented function(s)	5
The file does not contain a digital Certificate	7
The file references (3) whitelist strings	9
The file ignores Data Execution Prevention (DEP)	9
The file ignores Address Space Layout Randomization (ASLR)	9
The manifest identity name is "Microsoft.Windows.Shell.calc"	9
The file references a debug symbols file (path:"calc.pdb")	9
The file ignores cookies on the stack (GS)	9
The file ignores Code Integrity	9

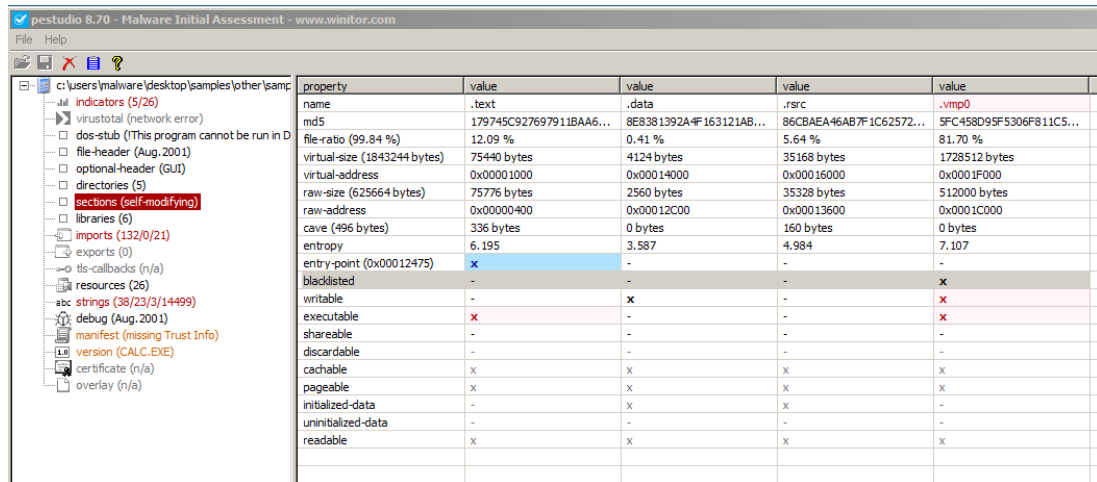
Figura 4: PESTudio - Indicators

Gli *indicatori* identificano una caratteristica o un comportamento del software. Ognuno di questi ha un valore numerico di severità che indica quanto esso è pericoloso in base a quanto viene ritenuta sospetta (un valore basso corrisponde ad un grado di sospetto alto).

Si nota che la presenza di una sezione *.vmp0* è segnalata come molto sospetta, così come un uso di funzioni e stringhe nella blacklist. Questa sezione risulta inoltre eseguibile e self-modificabile, e può modificare anche registri di altri eseguibili (come ad esempio di Windows Explorer).

Come si può vedere dalla *Figura 4*, ci sono 5 indicatori “rossi”, che hanno valore 1. Sono presenti anche altri indicatori con valore 2 (che il software rileva comunque come potenzialmente pericolosi) e altri ancora che non sono rilevati come malevoli o pericolosi.

## Analisi delle sezioni



property	value	value	value	value
name	.text	.data	.rsrc	.vmp0
md5	179745C927697911BAA6...	8E8381392A4F163121AB...	86CBAAE46AB7F1C62572...	5FC458D95F5306F811C5...
file-ratio (99.84 %)	12.09 %	0.41 %	5.64 %	81.70 %
virtual-size (1843244 bytes)	75440 bytes	4124 bytes	35168 bytes	1728512 bytes
virtual-address	0x00001000	0x00014000	0x00016000	0x0001F000
raw-size (625664 bytes)	75776 bytes	2560 bytes	35328 bytes	512000 bytes
raw-address	0x00000400	0x00012C00	0x00013600	0x0001C000
cave (496 bytes)	336 bytes	0 bytes	160 bytes	0 bytes
entropy	6.195	3.587	4.984	7.107
entry-point (0x00012475)	x	-	-	-
blacklisted	-	-	-	x
writable	-	x	-	x
executable	x	-	-	x
shareable	-	-	-	-
discardable	-	-	-	-
cacheable	x	x	x	x
pageable	x	x	x	x
initialized-data	-	x	x	-
uninitialized-data	-	-	-	-
readable	x	x	x	x

Figura 5: *PEStudio* - *Section*

Alla voce Sezioni del menù sono indicate le componenti dell'eseguibile. Le sezioni sono:

- **.text** contiene il codice eseguibile in chiaro. Probabilmente tale porzione di codice sarà destinata a decomprimere la parte offuscata.
- **.data** contiene variabili globali e variabili modificabili dal codice.
- **.rsrc** contiene varie risorse, tra cui immagini ed icone utilizzate.
- **.vmp0** è la sezione più importante e più offuscata del codice. Dal nome indica che è stata utilizzata dal software *VMPProtect*, un tool russo, per offuscare il codice così da rendere difficile il reverse-engineering.

La sezione *.vmp0* ha l'entropia maggiore (7.11), infatti anche il campo *"file-ratio"* che indica il rapporto di compressione del file è molto alto (81.70%). Tale dato avvalorava la nostra tesi. Questa sezione è probabilmente quella che eseguirà operazioni sospette, in quanto è l'unica con campo *"blacklist"* segnato, oltre al fatto che, insieme a *.text*, è anche eseguibile e scrivibile.

## Analisi delle librerie

pstudio 8.70 - Malware Initial Assessment - www.wintor.com

File Help

c:\users\malware\Desktop\samples\other\samp

indicators (5/26)

- virustotal (network error)
- dos-stub (!This program cannot be run in D
- file-header (Aug.2001)
- optional-header (GUI)
- directories (5)
- sections (self-modifying)
- libraries (6)
- imports (132/0/21)
- exports (0)
- tls-callbacks (n/a)
- resources (26)
- strings (38/23/3/14499)
- debug (Aug.2001)
- manifest (missing Trust Info)
- version (CALC.EXE)
- certificate (n/a)
- overlay (n/a)

library (6)	blacklist (0)	missing (0)	type	imports (132)	file-description
shell32.dll	-	-	Implicit	1	Windows Shell Common DLL
mshvrt.dll	-	-	Implicit	26	Windows NT CRT DLL
advapi32.dll	-	-	Implicit	3	Advanced Windows 32 Base API
kernel32.dll	-	-	Implicit	30	Windows NT BASE API Client DLL
gdi32.dll	-	-	Implicit	3	GDI Client DLL
user32.dll	-	-	Implicit	69	Multi-User Windows USER API Client DLL

Figura 6: *PEStudio - Librerie*

Le librerie presenti nel malware sono:

- **advapi32.dll** permette l'accesso a componenti avanzati di Windows (registri, service manager,...).
- **kernel32.dll** espone la maggior parte delle API di base di Win32 di tutte le applicazioni come gestione della memoria, operazioni di input / output, creazione di processi e thread e funzioni di sincronizzazione.
- **user32.dll** contiene tutti i componenti dell'interfaccia utente di Windows (bottoni, desktop, finestre,.. ), oltre a quelli per controllare e rispondere alle azioni dell'utente. Questo consente ai programmi di implementare un'interfaccia utente grafica (GUI) che si adatta al look and feel di Windows. I programmi chiamano funzioni da USER di Windows per eseguire operazioni come la creazione e la gestione di finestre, la ricezione di messaggi di finestre, la visualizzazione di testo in una finestra e la visualizzazione di messaggi.
- **gdi32.dll** le applicazioni chiamano direttamente le funzioni GDI (Graphic Device Interface) per eseguire disegni di basso livello (linea, rettangolo, ellisse), l'output del testo, la gestione dei font e funzioni simili. Viene inoltre utilizzata nella versione XP di Windows per Paint.  
Da semplici disegni, la funzionalità si è ampliata nel corso degli anni e ora include il supporto per caratteri TrueType, canali alfa e monitor multipli.
- **msvcrt.dll** alcuni malware possono sfruttarla per camuffarsi. Nel nostro caso l'eseguibile si maschera come la calcolatrice di Windows.
- **shell32.dll** usata per il sistema operativo Windows a 32 bit, consente alle applicazioni di accedere alle funzioni fornite dalla shell del sistema operativo per modificarle e migliorarle.

Sono quindi tutte librerie per Windows a 32 bit.

Abbiamo notato che non sono presenti librerie per connessione di rete, come *wsock32.dll*.



## Analisi degli import

symbol (132)	group (disabled)	blacklist (21)	anonymous (0)	anti-debug (0)	undocumented (1)	deprecated (0)	library (6)
GetModuleHandleA	-	x	-	-	-	-	kernel32.dll
LoadLibraryA	-	x	-	-	-	-	kernel32.dll
GetProcAddress	-	x	-	-	-	-	kernel32.dll
GlobalCompact	-	x	-	-	x	-	kernel32.dll
Sleep	-	x	-	-	-	-	kernel32.dll
WriteProfileStringW	-	x	-	-	-	-	kernel32.dll
GetStartupInfoA	-	x	-	-	-	-	kernel32.dll
CreateThread	-	x	-	-	-	-	kernel32.dll
GetCommandLineW	-	x	-	-	-	-	kernel32.dll
GetProfileIntW	-	x	-	-	-	-	kernel32.dll
CallWindowProcW	-	x	-	-	-	-	user32.dll
WinHelpW	-	x	-	-	-	-	user32.dll
PostQuitMessage	-	x	-	-	-	-	user32.dll
IsClipboardFormatAvailable	-	x	-	-	-	-	user32.dll
GetDesktopWindow	-	x	-	-	-	-	user32.dll
OpenClipboard	-	x	-	-	-	-	user32.dll
GetClipboardData	-	x	-	-	-	-	user32.dll
CloseClipboard	-	x	-	-	-	-	user32.dll
SendMessageW	-	x	-	-	-	-	user32.dll
SetWindowLongW	-	x	-	-	-	-	user32.dll
SystemParametersInfoW	-	x	-	-	-	-	user32.dll
ShellAboutW	-	-	-	-	-	-	shell32.dll
_CoxFrameHandler	-	-	-	-	-	-	msvart.dll
_CoxThrowException	-	-	-	-	-	-	msvart.dll
wcsnlen	-	-	-	-	-	-	msvart.dll
toupper	-	-	-	-	-	-	msvart.dll
wcschr	-	-	-	-	-	-	msvart.dll
memmove	-	-	-	-	-	-	msvart.dll
wcslen	-	-	-	-	-	-	msvart.dll
_wcsrev	-	-	-	-	-	-	msvart.dll
_c_exit	-	-	-	-	-	-	msvart.dll
_exit	-	-	-	-	-	-	msvart.dll
_xcpFilter	-	-	-	-	-	-	msvart.dll
_cexit	-	-	-	-	-	-	msvart.dll
_exit	-	-	-	-	-	-	msvart.dll
_acordn	-	-	-	-	-	-	msvart.dll
_getmainargs	-	-	-	-	-	-	msvart.dll
_initterm	-	-	-	-	-	-	msvart.dll
_setusermatherr	-	-	-	-	-	-	msvart.dll

sha256: 34558AC3FAB17CA1A1FF70860B35296395F1DF7FABD66839C56FAECF9C30FFC | cpu: 32-bit | file-type: executable | subsystem: GUI | entry-point: 0x0012475 | signature: n/a

Figura 7: PESTudio - Imports

Nel menù *import* si nota che le librerie *kernel32.dll* e *user32.dll* fanno parte della blacklist. Gli import più interessanti sono:

- **GetCommanLine**: restituisce un puntatore alla stringa della chiamata del programma. Legge il contenuto da riga di comando.
- **GetProcAddress**: prende l'indirizzo virtuale del chiamante.
- **CreateThread** e **Sleep**: sono spesso usati in combinazione per evitare che il virus si ostacoli da solo quando si istanziano più chiamate del *sample2.exe*. Ovviamente queste due chiamate di sistema sono invocate tramite l'utilizzo di semafori.

Tra le operazioni non segnate nella blacklist ci sono *RegOpenKeyExA*, *RegQueryValueExA*, *RegCloseKey* che consentono al virus di modificare le chiavi dei registri. Tali operazioni sono comunque malevole se pur non considerate da Windows Defender come tali.

La ragione per la quale Windows Defender non le riconosce come tali è per il fatto che sono normali operazioni di lettura e scrittura dei registri. Queste 3 chiamate di sistema (*RegOpenKeyExA*, *RegQueryValueE-xa*, *RegCloseKey*) sono usate dal virus per modificare le chiavi di registro (ad esempio *Internet Explorer* non si riesce più ad eseguire dopo la modifica). Ci sono anche altri import come *CallWindowsProcess*, *Open* e *Close Clipboard*, *WinHelpW* e altre.

Per concludere si osserva che tutte queste chiamate sono piuttosto inusuali (come la manipolazione dei registri e la creazione di processi) dato che l'eseguibile dovrebbe essere una normale calcolatrice.

## Analisi delle stringhe

type	size	location	blacklist (38)	hint (23)	whitelist (3)	value (14499)
ascii	10	0x0000...	x	-	-	hhctrl.ocx
ascii	11	.text:0...	x	-	-	RegCloseKey
ascii	15	.text:0...	x	-	-	RegQueryValueEx
ascii	12	.text:0...	x	-	-	RegOpenKeyEx
ascii	14	.text:0...	x	-	-	GetCommandLine
ascii	10	.text:0...	x	-	-	LocalAlloc
ascii	16	.text:0...	x	-	-	GetProfileString
ascii	13	.text:0...	x	-	-	GetProfileInt
ascii	12	.text:0...	x	-	-	LocalReAlloc
ascii	8	.text:0...	x	-	-	SetEvent
ascii	10	.text:0...	x	-	-	ResetEvent
ascii	12	.text:0...	x	-	-	CreateThread
ascii	12	.text:0...	x	-	-	GlobalUnlock
ascii	10	.text:0...	x	-	-	GlobalSize
ascii	10	.text:0...	x	-	-	GlobalLock
ascii	18	.text:0...	x	-	-	WriteProfileString
ascii	5	.text:0...	x	-	-	Sleep
ascii	13	.text:0...	x	-	-	GlobalReAlloc
ascii	10	.text:0...	x	-	-	GlobalFree
ascii	11	.text:0...	x	-	-	GlobalAlloc
ascii	13	.text:0...	x	-	-	GlobalCompact
ascii	14	.text:0...	x	-	-	GetProcAddress
ascii	11	.text:0...	x	-	-	LoadLibrary
ascii	15	.text:0...	x	-	-	GetModuleHandle
ascii	14	.text:0...	x	-	-	GetStartupInfo
ascii	11	.text:0...	x	-	-	SendMessage
ascii	13	.text:0...	x	-	-	SetWindowLong
ascii	20	.text:0...	x	-	-	SystemParametersInfo
ascii	14	.text:0...	x	-	-	CloseClipboard
ascii	16	.text:0...	x	-	-	GetClipboardData
ascii	13	.text:0...	x	-	-	OpenClipboard
ascii	16	.text:0...	x	-	-	GetDesktopWindow
ascii	26	.text:0...	x	-	-	IsClipboardFormatAvailable
ascii	15	.text:0...	x	-	-	PostQuitMessage
ascii	7	.text:0...	x	-	-	WinHelp
ascii	14	.text:0...	x	-	-	CallWindowProc
unicode	15	.rsrc:0...	x	-	-	FileDescription
unicode	11	.rsrc:0...	x	-	-	FileVersion
ascii	40	0x0000...	-	x	-	!This program cannot be run in DOS mode.

sha256: 34558AC38FAB17CA1A1FF70860B35296399F1DF7FA8D86B39C56FAECF9C3CFFC | cpu: 32-bit | file-type: executable | subsystem: GUI | entry-point: 0x00012475

Figura 8: PESTudio - Strings

In questa sezione troviamo le *stringhe* con le relative chiamate a funzione che sono presenti nei vari import analizzati prima.

Analizzando le sezioni *.text*, *.data* e *.rsrc* notiamo che sono tutte in chiaro.

Tra le operazioni eseguite, le più importanti sono le modifiche dei registri già indicate sopra.

Oltre alle chiamate rilevate e blacklisted ci sono tutte le altre chiamate che sono offuscate e quindi non rilevabili.

Le chiamate offuscate appartengono, quasi sicuramente, alla sezione *.vmp0*.

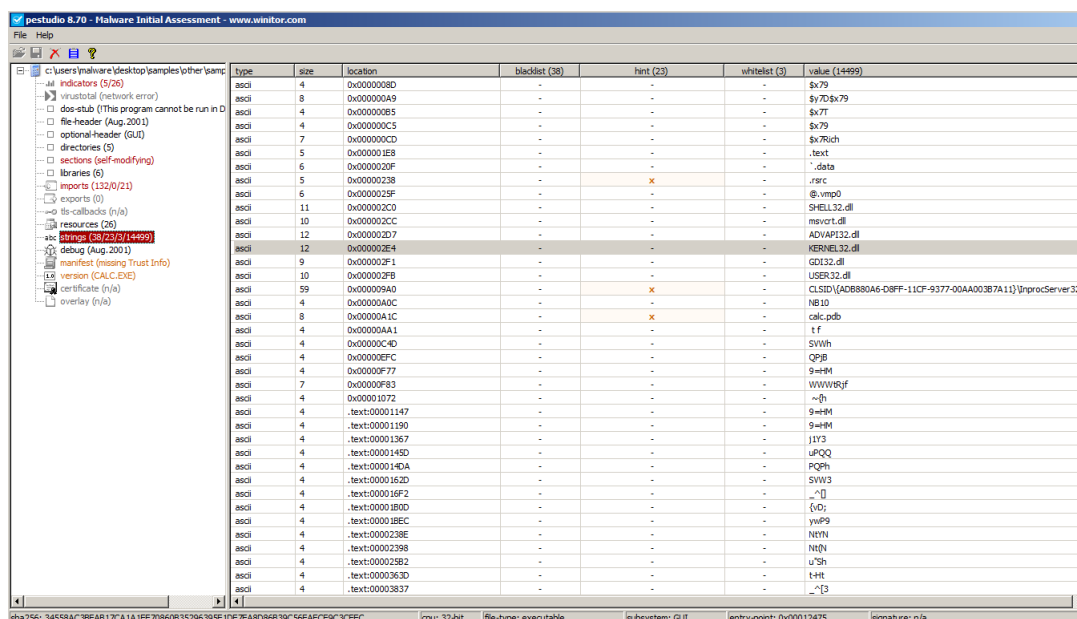


Figura 9: *PEStudio - Strings Offuscated*

## Analisi con PEID

Il tool PEID, già presente in *Windows.ova*, sta per Packer Identification Tools.

Esso ha tre modalità di analisi:

- **Normal Mode** : esegue la scansione dei file eseguibili nel loro punto di ingresso per tutte le firme documentate. Questo è ciò che fanno anche tutti gli altri identificatori.
- **Deep Mode** : esegue la scansione del punto di ingresso del file eseguibile contenente la sezione per tutte le firme documentate. Ciò garantisce il rilevamento di circa l'80% dei file modificati e criptati.
- **Hardcore Mode** : esegue una scansione completa dell'intero file eseguibile per le firme documentate. Si dovrebbe usare questa modalità come ultima opzione poiché le firme piccole tendono spesso a presentarsi molto in molti file e quindi potrebbero restituire risultati errati, generando dei falsi positivi.

I primi due metodi restituiscono output quasi istantanei, ma l'ultimo metodo è più lento per ovvi motivi.

Da una prima analisi, quindi in *Normal Mode*, l'eseguibile risulterebbe essere packed, mentre un'analisi con *Hardcore Mode* risulta essere non packed. (N.B. Con **virus packed** si intende un *virus che è stato compresso e che quindi deve essere decompresso a run time, in una singola esecuzione*).

In conclusione possiamo dire che il virus non è packed, come detto nell'*Hardcore Mode*. Non possiamo dire tanto altro e non siamo sicuri che questa valutazione sia corretta, in quanto il tool *PEID* è obsoleto.

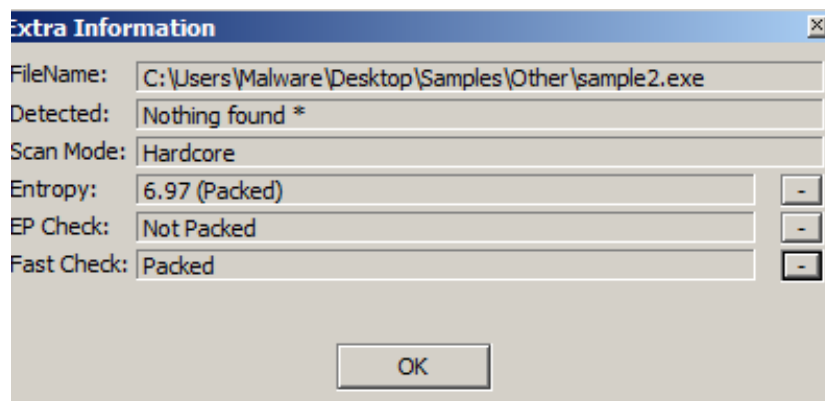


Figura 10: *Peid - malware not packed*

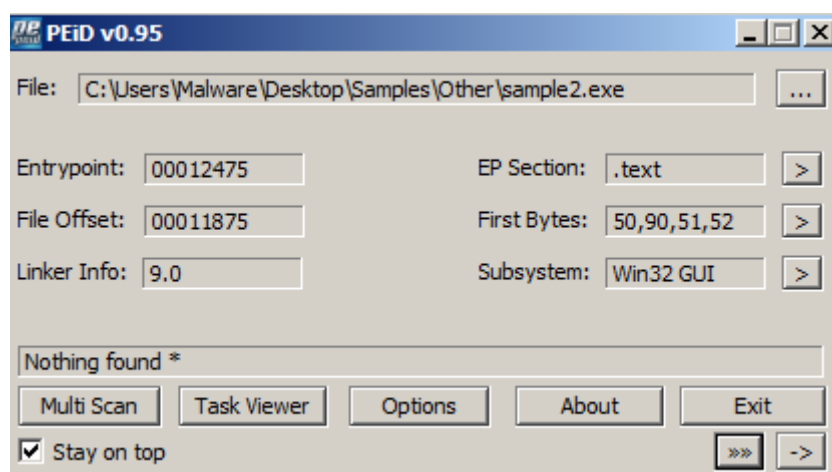


Figura 11: *Peid - User Interface*

## VirusTotal

Inserendo sul sito *VirusTotal.com* l'*hash md5* si possono ottenere tutte le informazioni del virus che sono presenti online, in quanto già analizzato da utenti e anti-virus, conoscendone quindi a pieno le sue peculiarità e la tipologia.

Queste informazioni sono una ripetizione di quelle che abbiamo appena trovato con *PEStudio*.

Dal sito si evidenzia la forte somiglianza con la classe di virus *Expiro*.

Tale classe di malware generalmente esegue le seguenti azioni:

- *Modificare numerose chiavi di registro*
- *Modifica di file*
- *Infezione di eseguibili, tipico di un comportamento polimorfico.*
- *Sincronismo con mutex*
- *Furto di dati ed invio all'esterno con connessioni HTTP*

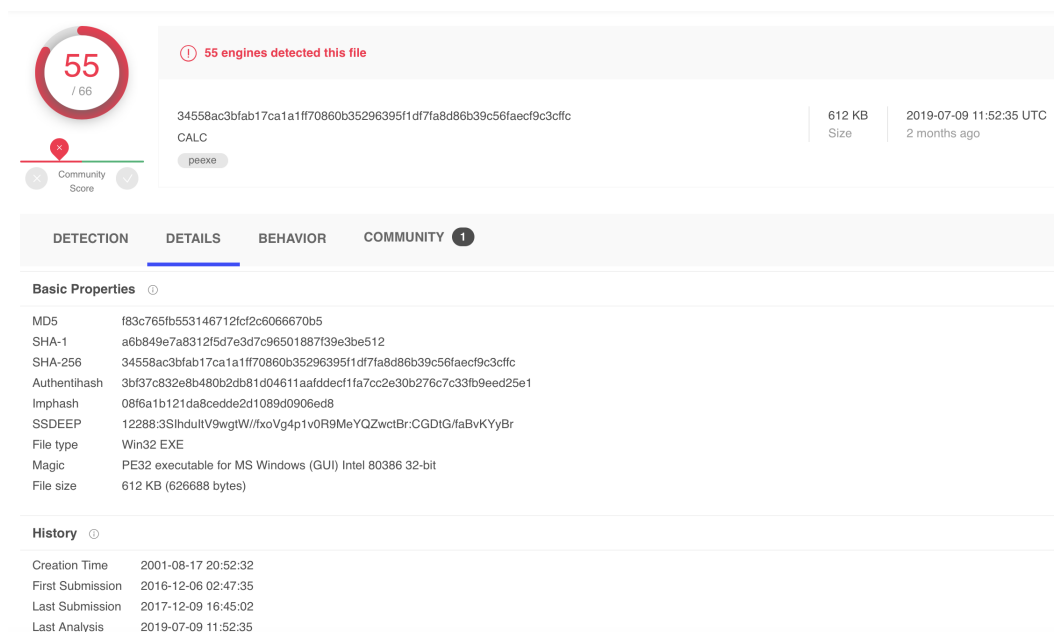


Figura 12: *VirusTotal Details*

Il punteggio dato dal sito al file analizzato è 55 su 66, quindi è rilevato come *file malevolo* circa nell'83% delle volte.

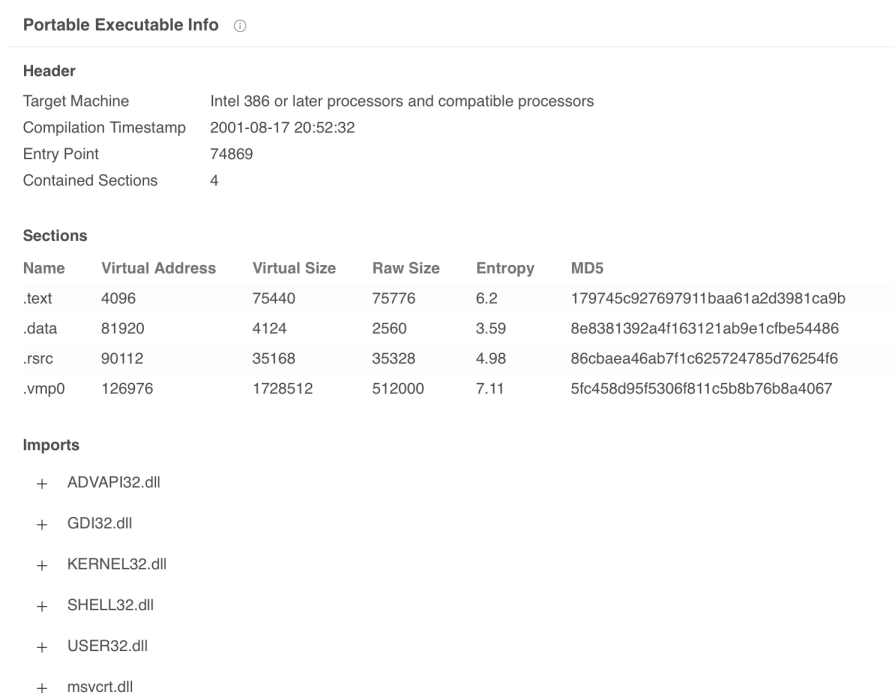


Figura 13: *VirusTotal - Header, Section, Imports*

Qui vengono mostrate le informazioni relative al virus, al suo header, al tipo (legato al fatto che si "maschera" come una calcolatrice di Windows) e altri parametri che abbiamo già visto con *PEStudio* ma raggruppati.

Come detto prima, i comportamenti sono molteplici e sono visibili su VirusTotal.

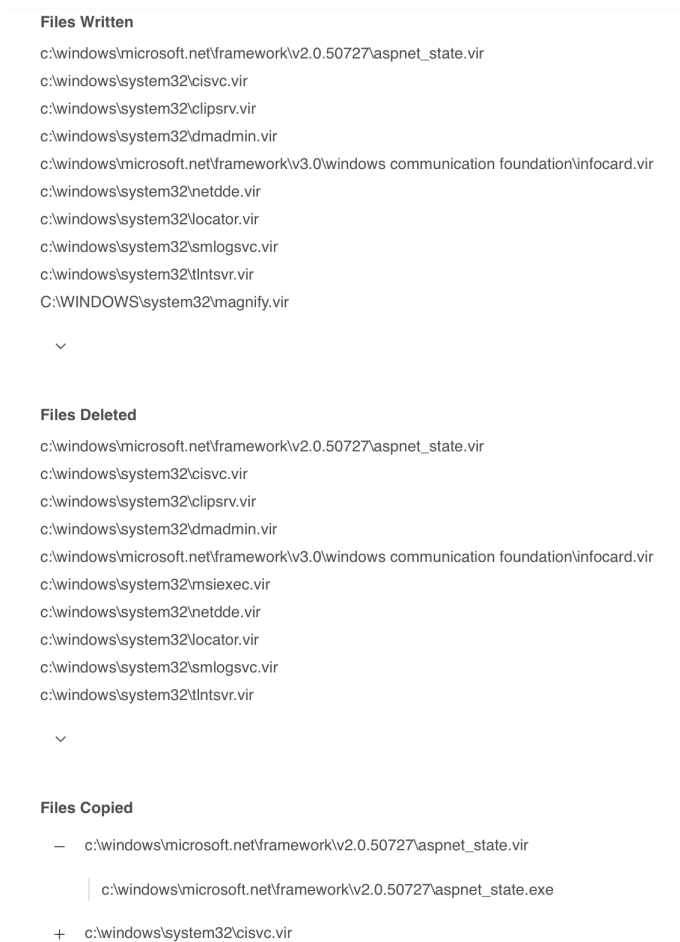


Figura 14: *VirusTotal - Behavior*

Il virus elimina file *.vir*, che solitamente sono associati a file modificati dal virus e che vengono riconosciuti come malevoli dall'antivirus installato sul PC. Il programma antivirus, al momento del rilevamento del virus, modifica l'estensione in *.vir* per identificare tale file come infetto e proteggere l'utente impedendone l'apertura.

## Parte II

# Analisi Dinamica

*L'analisi dinamica consiste nell'osservare le funzionalità del file in esame dal "vivo". Di solito l'analisi dinamica viene eseguita dopo quella statica. Si segue questo schema perchè una stringa eseguibile in analisi statica potrebbe essere non eseguita dal programma stesso.*

Anche per la parte di analisi dinamica usiamo la stessa macchina virtuale con sistema operativo Windows 7 e dei tool per questa particolare analisi che sono già presenti in *Windows.ova*. Per eseguire questa parte abbiamo bisogno anche della *Fakenet*, con la quale si simula una connessione verso l'esterno in modo da far credere al virus di aver accesso ad Internet a tutti i programmi che la richiedono.

L'analisi è stata effettuata a partire da una snapshot della macchina virtuale Windows 7, di volta in volta ricreata a partire da quella non infetta, in modo da poter analizzare più volte e in totale sicurezza il comportamento del malware senza recare danni alla propria macchina.

Come, già detto prima, l'eseguibile si presenta come una calcolatrice con la GUI di Windows. Mostriamo ora, con un'immagine, come si presenta l'eseguibile malevolo.

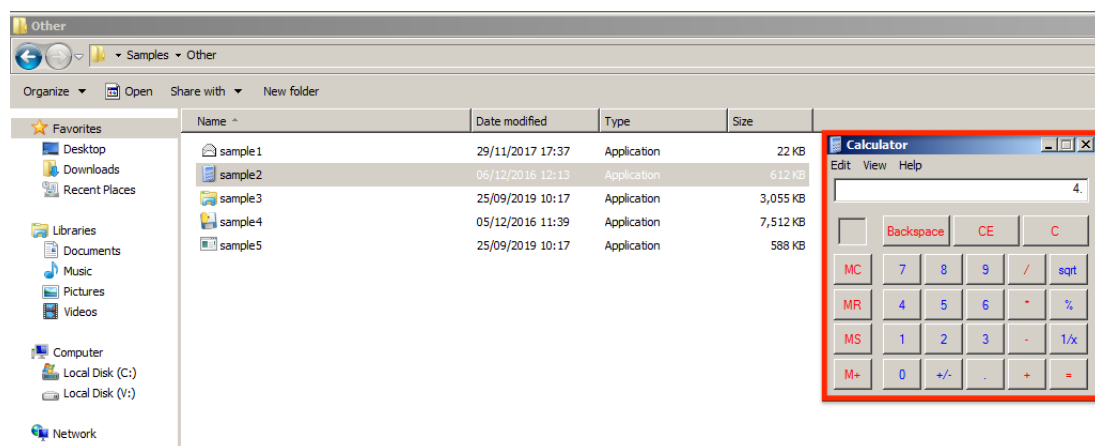


Figura 15: *Windows - Malware execution*

## Expiro

Il comportamento atteso di tale malware, come già anticipato, è di tipo Expiro, il quale può funzionare su sistemi operativi a 32 e 64 bit.

Quindi ci aspettiamo:

- Installazione di estensioni malevole
- Rollback delle specifiche di sicurezza (Es. disabilitazione di Windows Defender)
- Phishing di credenziali ed invio in rete

Il virus è altamente infettivo e la modalità di infezione si articola in due fasi:

- **Fase I** : Copia del programma modificando l'estensione da *.exe* a *.vir*.

- **Fase II:** Aggiunge in coda il codice (ovvero *.vmp0*). Ripristina il file *.exe*, disattivando tal volta *Windows File Protection*.

Un esempio è la modifica dell'eseguibile *alg.exe* rilevata tramite tool *ProcMon*. *alg.exe* è un servizio Application Layer Gateway fa parte delle funzionalità di Condivisione della Connessione Internet (ICS) e del firewall di Windows su Windows XP. Permette alle applicazioni come i client di messaging e i programmi per il trasferimento dei file di usare le porte passive TCP/UDP per comunicare con un server.

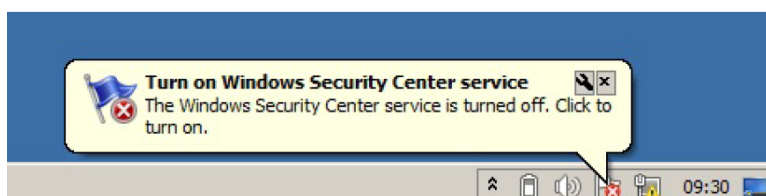


Figura 16: *Windows Defender - Notification*

Inoltre abbiamo notato, che all'avvio del malware, c'è la disabilitazione immediata del *Windows Security Center*. Tale comportamento è tipico di un malware che vuole abbassare le misure di sicurezza e vuole eseguire modifiche ai file.

Anticipiamo già che una volta lanciato il malware, in un breve tempo, si perdono i privilegi di esecuzione su molti eseguibili della macchina (come accennato prima con Internet Explorer).

14:11...	sample2.exe	4088	SetSecurityFile	C:\Windows\System32\alg.exe	SUCCESS	Information: DACL
14:11...	sample2.exe	4088	CloseFile	C:\Windows\System32\alg.exe	SUCCESS	
14:11...	sample2.exe	4088	CreateFile	C:\Windows\System32\alg.exe	SUCCESS	Desired Access: G...
14:11...	sample2.exe	4088	QueryStandard...	C:\Windows\System32\alg.exe	SUCCESS	AllocationSize: 61...
14:11...	sample2.exe	4088	ReadFile	C:\Windows\System32\alg.exe	SUCCESS	Offset: 0, Length: 5...
14:11...	sample2.exe	4088	ReadFile	C:\Windows\System32\alg.exe	SUCCESS	Offset: 0, Length: 5...
14:11...	sample2.exe	4088	CloseFile	C:\Windows\System32\alg.exe	SUCCESS	
14:11...	sample2.exe	4088	CreateFile	C:\Windows\System32\alg.vir	SUCCESS	Desired Access: G...
14:11...	sample2.exe	4088	WriteFile	C:\Windows\System32\alg.vir	SUCCESS	Offset: 0, Length: 5...
14:11...	sample2.exe	4088	CloseFile	C:\Windows\System32\alg.vir	SUCCESS	
14:11...	sample2.exe	4088	RegOpenKey	HKLM\Software\Policies\Microsoft\Windows\System	SUCCESS	Desired Access: Q...
14:11...	sample2.exe	4088	RegQueryValue	HKLM\SOFTWARE\Policies\Microsoft\Windows\System\CopyFileBufferedSynchronousIo	NAME NOT FOUND	Length: 20
14:11...	sample2.exe	4088	RegCloseKey	HKLM\SOFTWARE\Policies\Microsoft\Windows\System	SUCCESS	

Figura 17: *Procmon - Esempio su alg.exe*

La chiave di registro (*wscsvc*) che si modifica è:

**HKLM\SYSTEM\CurrentControlSet\services\wscsvs\Start: 0x00000002**

**HKLM\SYSTEM\CurrentControlSet\services\wscsvc\Start: 0x00000004**

Figura 18: *Windows Defender - Key*

Analizzando le cartelle nascoste in *C:\Users\Malware\AppData\Local*, il malware ha creato un libreria tipica di questa tipologia di virus, che servirà per salvare le informazioni rubate. Il nome di tale libreria (*usr28zt32.dll*) è molto conosciuta in letteratura, perchè se ricercata online rimanda subito ad una tipologia di virus *Expiro* piuttosto famosa negli anni '90.



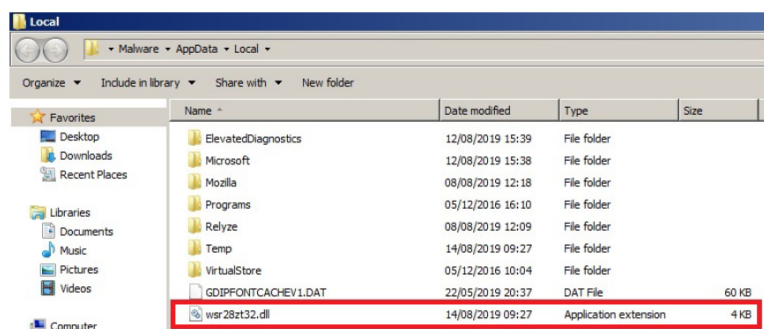


Figura 19: *Expiro* - Library created

## Analisi tramite RegShot e ProcMon

Presentato l'ambiente di testing nel quale sono state fatte le analisi comportamentali del malware, passiamo ora ad elencare le caratteristiche riscontrate.

I passi eseguiti ad ogni run del virus sono stati i seguenti e nel seguente ordine preciso:

1. Avvio di Fakenet
2. Avvio **Procmon** con permessi di amministrazione, reset e settaggio dei filtri desiderati
3. Avvio **Regshot** definendo il percorso di analisi ed esecuzione della prima cattura dello stato dei registri
4. Avvio rilevazione di Procmon e run del virus
5. Interazione con la GUI del virus eseguibile
6. Chiusura della GUI del presunto malware (*calcolatrice*)
7. Stop di Procmon e seconda cattura dello stato dei registri
8. Confronto dei risultati delle due catture
9. Analisi del comportamento di fakenet

L'eseguibile non dà subito segnali di attacco all'utente, infatti provando ad eseguire un semplice calcolo esso ritorna il risultato desiderato e non si manifestano anomalie nel sistema. Dopo alcuni minuti l'infezione si manifesta visibilmente osservando le icone di alcuni file eseguibili (come ad esempio *Internet Explorer*), che sono contrassegnate da un lucchetto, sinonimo di modifica dei permessi e delle proprietà.

L'infezione si articola partendo dalla copia della sezione *.vmp0* del presunto malware all'interno dei file infetti. La sezione *.vmp0* presenta grandi dimensioni ed entropia elevata, sinonimo di tecniche di offuscamento del codice. Per trasferire il controllo al corpo principale (*.vmp0*), il presunto virus inserisce 1248 bytes di codice di avvio malevolo al posto del punto di ingresso (entry point). Questo codice di avvio esegue la decompressione del codice del virus nella sezione offuscata.

Per mostrare l'infezione da parte del malware, prendiamo come esempio *Internet Explorer*.

property	value	value	value	value	value	value	value
name	.text	.data	.idata	.rsrc	.reloc		
md5	79FD289C8B53CE13504B...	6156079D89C15F6CD4D5...	E7382D06FA9E13D48C3A...	7500089F925DCBD7820...	55348021002E70AFC64...	93902E0E801D9143893...	
file-ratio (97.97 %)	2.07 %	0.19 %	0.31 %	0.06 %	95.20 %	0.13 %	
virtual-size (797041 bytes)	16765 bytes	1264 bytes	2168 bytes	76 bytes	775744 bytes	1024 bytes	
optional-header (GUI)	0x00001000	0x00006000	0x00007000	0x00008000	0x00009000	0x000C7000	
raw-size (798720 bytes)	16896 bytes	1536 bytes	2560 bytes	512 bytes	776192 bytes	1024 bytes	
raw-address	0x00000400	0x00004600	0x00004C00	0x00005600	0x00005800	0x000C3000	
case (1479 bytes)	131 bytes	272 bytes	392 bytes	436 bytes	448 bytes	0 bytes	
entropy	6.164	0.182	4.586	0.682	6.461	6.537	
entry-point (0x00001E40)	x	-	-	-	-	-	
blacklisted	-	-	-	x	-	-	
writable	-	x	-	x	-	-	
executable	x	-	-	-	-	-	
shareable	-	-	-	-	-	-	
discardable	-	-	-	-	-	-	
cacheable	x	x	x	x	x	x	
pageable	x	x	x	x	x	x	
initialized-data	-	x	x	x	x	x	
uninitialized-data	-	-	-	-	-	-	
readable	x	x	x	x	x	x	

Figura 20: Internet Explorer - Before infection

property	value	value	value	value	value	value	value
name	.text	.data	.idata	.ddat	.rsrc	.reloc	.vmp0
md5	5A999445504364390E78...	6156079D89C15F6CD4D5...	E7382D06FA9E13D48C3A...	7500089F925DCBD7820...	55348021002E70AFC64...	93902E0E801D9143893...	8F456CEAF82B19E780B...
file-ratio (99.92 %)	1.29 %	0.12 %	0.20 %	0.04 %	59.17 %	0.88 %	39.03 %
virtual-size (525553 bytes)	16765 bytes	1264 bytes	2168 bytes	76 bytes	775744 bytes	1728512 bytes	1728512 bytes
optional-header (GUI)	0x00001000	0x00006000	0x00007000	0x00008000	0x00009000	0x000C7000	0x000C3000
raw-size (1310720 bytes)	16896 bytes	1536 bytes	2560 bytes	512 bytes	776192 bytes	512000 bytes	512000 bytes
raw-address	0x00000400	0x00004600	0x00004C00	0x00005600	0x00005800	0x000C3000	0x000C3400
case (1479 bytes)	131 bytes	272 bytes	392 bytes	436 bytes	448 bytes	0 bytes	0 bytes
entropy	6.211	0.182	4.586	0.682	6.461	6.537	7.095
entry-point (0x00001E40)	x	-	-	-	-	-	-
blacklisted	-	-	-	x	-	-	x
writable	-	x	-	x	-	-	x
executable	x	-	-	-	-	-	x
shareable	-	-	-	-	-	-	-
discardable	-	-	-	-	-	-	-
cacheable	x	x	x	x	x	x	x
pageable	x	x	x	x	x	x	x
initialized-data	-	x	x	x	x	-	-
uninitialized-data	-	-	-	-	-	-	-
readable	x	x	x	x	x	x	x

Figura 21: Internet Explorer - After infection

Come si vede dalle immagini, si ha l'inserimento di una nuova sezione, `.vmp0`, che porta ad infettare l'eseguibile.

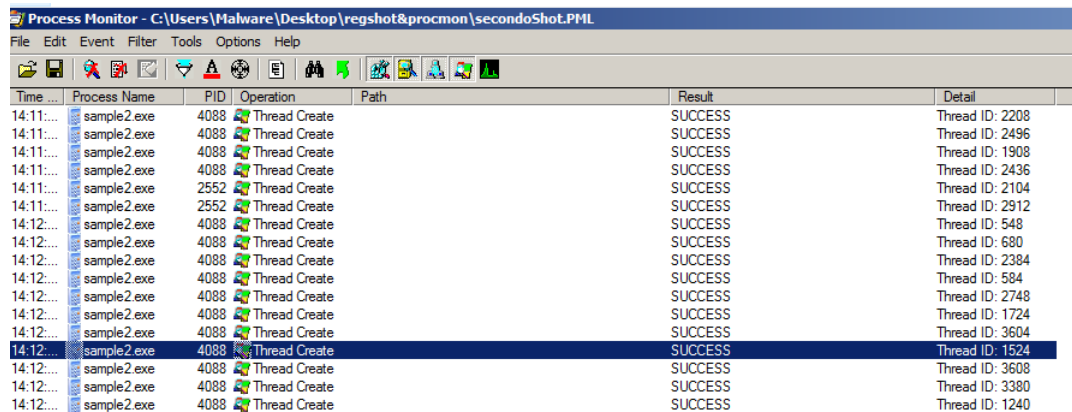
Il virus tenta di modificare le seguenti chiavi di sistema:

- HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\ZoneNum
- HKCU\Software\Microsoft\Internet Explorer\IntelliForms\Storage
- HKLM\System\CurrentControlSet\Control\ComputerName\ActivateComputerName

Time	Process Name	PID	Operation	Path	Result	Detail
14.12.	sample2.exe	4088	RegSet Value	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\0.1609	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
14.12.	sample2.exe	4088	RegSet Value	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\0.1406	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
14.12.	sample2.exe	4088	RegSet Value	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\0.2103	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
14.12.	sample2.exe	4088	RegSet Value	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\0.1169	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
14.12.	sample2.exe	4088	RegSet Value	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\0.1146	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
14.12.	sample2.exe	4088	RegSet Value	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\1.1609	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
14.12.	sample2.exe	4088	RegSet Value	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\2.1609	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
14.12.	sample2.exe	4088	RegSet Value	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\2.1406	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
14.12.	sample2.exe	4088	RegSet Value	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\2.2103	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
14.12.	sample2.exe	4088	RegSet Value	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\3.1609	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
14.12.	sample2.exe	4088	RegSet Value	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\3.1406	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
14.12.	sample2.exe	4088	RegSet Value	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\3.2103	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
14.12.	sample2.exe	4088	RegSet Value	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\4.1609	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
14.12.	sample2.exe	4088	RegSet Value	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\4.1406	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
14.12.	sample2.exe	4088	RegSet Value	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\4.2103	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
14.12.	sample2.exe	4088	RegSet Value	HKCU\Software\Classes\Local Settings\MuCache\3C52C6487E\LanguageList	SUCCESS	Type: REG_MULTI_SZ, Length: 20, Data: en-US, en
14.12.	sample2.exe	4088	RegSet Value	HKCU\Software\Classes\Local Settings\MuCache\3C52C6487E\LanguageList	SUCCESS	Type: REG_MULTI_SZ, Length: 20, Data: en-US, en
14.12.	sample2.exe	4088	RegSet Value	HKCU\Software\Classes\Local Settings\MuCache\3C52C6487E\LanguageList	SUCCESS	Type: REG_MULTI_SZ, Length: 20, Data: en-US, en
14.12.	sample2.exe	4088	RegSet Value	HKCU\Software\Classes\Local Settings\MuCache\3C52C6487E\LanguageList	SUCCESS	Type: REG_MULTI_SZ, Length: 20, Data: en-US, en
14.12.	sample2.exe	4088	RegSet Value	HKCU\Software\Classes\Local Settings\MuCache\3C52C6487E\LanguageList	SUCCESS	Type: REG_MULTI_SZ, Length: 20, Data: en-US, en
14.12.	sample2.exe	4088	RegSet Value	HKCU\Software\Classes\Local Settings\MuCache\3C52C6487E\LanguageList	SUCCESS	Type: REG_MULTI_SZ, Length: 20, Data: en-US, en
14.12.	sample2.exe	4088	RegSet Value	HKCU\Software\Classes\Local Settings\MuCache\3C52C6487E\LanguageList	SUCCESS	Type: REG_MULTI_SZ, Length: 20, Data: en-US, en
14.12.	sample2.exe	4088	RegSet Value	HKCU\Software\Classes\Local Settings\MuCache\3C52C6487E\LanguageList	SUCCESS	Type: REG_MULTI_SZ, Length: 20, Data: en-US, en
14.12.	sample2.exe	4088	RegSet Value	HKCU\Software\Classes\Local Settings\MuCache\3C52C6487E\LanguageList	SUCCESS	Type: REG_MULTI_SZ, Length: 20, Data: en-US, en
14.12.	sample2.exe	4088	RegSet Value	HKCU\Software\Classes\Local Settings\MuCache\3C52C6487E\LanguageList	SUCCESS	Type: REG_MULTI_SZ, Length: 20, Data: en-US, en
14.12.	sample2.exe	4088	RegSet Value	HKCU\Software\Classes\Local Settings\MuCache\3C52C6487E\LanguageList	SUCCESS	Type: REG_MULTI_SZ, Length: 20, Data: en-US, en

Figura 22: RegShot - Edit internet explorer setup logs

Infine, sapendo che il virus è altamente infettivo, esso genera molte copie di sè stesso, causando un'elevata densità di thread nei processi di sistema. La replicazione sembra essere binaria, ovvero ogni thread genera due copie figlie, oltre ad eseguire altre operazioni. Nonostante questo, il virus garantisce la sua mutua esclusione tramite l'uso dei semafori, della forma: `kkq-vx_mtx(numeroRandom)`.



The screenshot shows the Process Monitor application window with the title bar 'Process Monitor - C:\Users\Malware\Desktop\regshot&procmon\secondoShot.PML'. The interface includes a menu bar (File, Edit, Event, Filter, Tools, Options, Help) and a toolbar with various icons. The main display area is a table with columns: Time, Process Name, PID, Operation, Path, Result, and Detail. The table lists multiple 'Thread Create' events for 'sample2.exe' with PID 4088, all resulting in 'SUCCESS'. The thread IDs listed in the 'Detail' column are: 2208, 2496, 1908, 2436, 2104, 2912, 548, 680, 2384, 584, 2748, 1724, 3604, 1524, 3608, 3380, and 1240.

Time	Process Name	PID	Operation	Path	Result	Detail
14:11:...	sample2.exe	4088	Thread Create		SUCCESS	Thread ID: 2208
14:11:...	sample2.exe	4088	Thread Create		SUCCESS	Thread ID: 2496
14:11:...	sample2.exe	4088	Thread Create		SUCCESS	Thread ID: 1908
14:11:...	sample2.exe	4088	Thread Create		SUCCESS	Thread ID: 2436
14:11:...	sample2.exe	2552	Thread Create		SUCCESS	Thread ID: 2104
14:11:...	sample2.exe	2552	Thread Create		SUCCESS	Thread ID: 2912
14:12:...	sample2.exe	4088	Thread Create		SUCCESS	Thread ID: 548
14:12:...	sample2.exe	4088	Thread Create		SUCCESS	Thread ID: 680
14:12:...	sample2.exe	4088	Thread Create		SUCCESS	Thread ID: 2384
14:12:...	sample2.exe	4088	Thread Create		SUCCESS	Thread ID: 584
14:12:...	sample2.exe	4088	Thread Create		SUCCESS	Thread ID: 2748
14:12:...	sample2.exe	4088	Thread Create		SUCCESS	Thread ID: 1724
14:12:...	sample2.exe	4088	Thread Create		SUCCESS	Thread ID: 3604
14:12:...	sample2.exe	4088	Thread Create		SUCCESS	Thread ID: 1524
14:12:...	sample2.exe	4088	Thread Create		SUCCESS	Thread ID: 3608
14:12:...	sample2.exe	4088	Thread Create		SUCCESS	Thread ID: 3380
14:12:...	sample2.exe	4088	Thread Create		SUCCESS	Thread ID: 1240

Figura 23: Thread - Example of created threads

## Analisi tramite FakeNet e Wireshark

### FakeNet

Il programma malevolo, dopo aver collezionato le informazioni riguardanti versione di Windows, versione di Internet Explorer, avendole raccolte nella libreria citata all'inizio (*wsr28zt32.dll*), tenta di inviarle utilizzando l'eseguibile *dllhost.exe*, interno al virus (lo stesso del sistema operativo Windows e di conseguenza non rilevabile da Windows Defender come minaccia)

*dllhost.exe* viene sfruttato per effettuare operazioni *POST HTTP* periodiche ad un elenco di domini web malevoli generato randomicamente tramite algoritmi appositi denominati *DGA* (Domain Generation Algorithm), inviando alcune informazioni di sistema come di seguito si può notare.

Nella seguente immagine si possono notare tutti i file di testo contenenti gli indirizzi HTTP intercettati da FakeNet.

listeners	28/11/2017 15:42	File folder	
CHANGELOG	28/11/2017 15:42	Text Document	1 KB
FM fakenet	28/11/2017 15:42	Application	6,852 KB
fakenet.exe.manifest	28/11/2017 15:42	MANIFEST File	1 KB
http_20190919_091342	19/09/2019 11:57	Text Document	1 KB
http_20190919_091343	19/09/2019 11:57	Text Document	1 KB
http_20190919_091344	19/09/2019 11:57	Text Document	1 KB
http_20190919_091345	19/09/2019 11:57	Text Document	1 KB
http_20190919_091346	19/09/2019 11:57	Text Document	1 KB
http_20190919_091347	19/09/2019 11:57	Text Document	1 KB
http_20190919_091348	19/09/2019 11:57	Text Document	1 KB
http_20190919_091349	19/09/2019 11:57	Text Document	1 KB
http_20190919_091351	19/09/2019 11:57	Text Document	1 KB
http_20190919_091352	19/09/2019 11:57	Text Document	1 KB
http_20190919_091353	19/09/2019 11:57	Text Document	1 KB
http_20190919_091355	19/09/2019 11:57	Text Document	1 KB
http_20190919_091356	19/09/2019 11:57	Text Document	1 KB
http_20190919_091357	19/09/2019 11:57	Text Document	1 KB
http_20190919_091358	19/09/2019 11:57	Text Document	1 KB
http_20190919_091359	19/09/2019 11:57	Text Document	1 KB
http_20190919_091400	19/09/2019 11:57	Text Document	1 KB
http_20190919_091401	19/09/2019 11:57	Text Document	1 KB
http_20190919_091402	19/09/2019 11:57	Text Document	1 KB

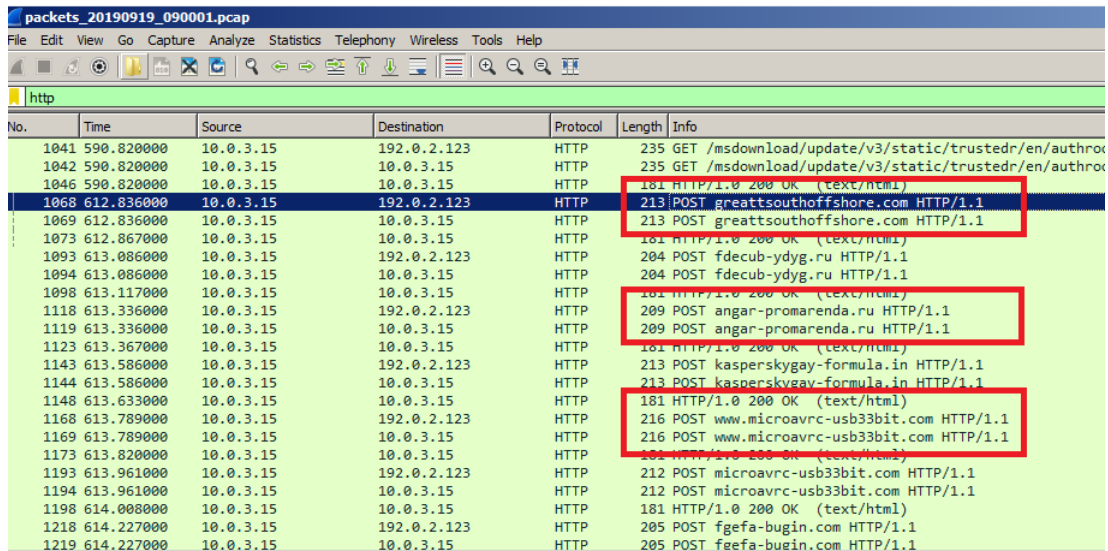
Figura 24: *FakeNet - Intercepted HTTP addresses*

Questi indirizzi internet sono presenti nel terminale: il virus cerca di comunicare con gli URL per inviare le informazioni carpite dal sistema.

Questi siti sono per la maggior parte russi (ad oggi sono tutti non raggiungibili), ciò è indicato dal fatto che il virus è stato creato nel 2001 e ad oggi è già stato analizzato da anti-virus e vari utenti.

# Wireshark

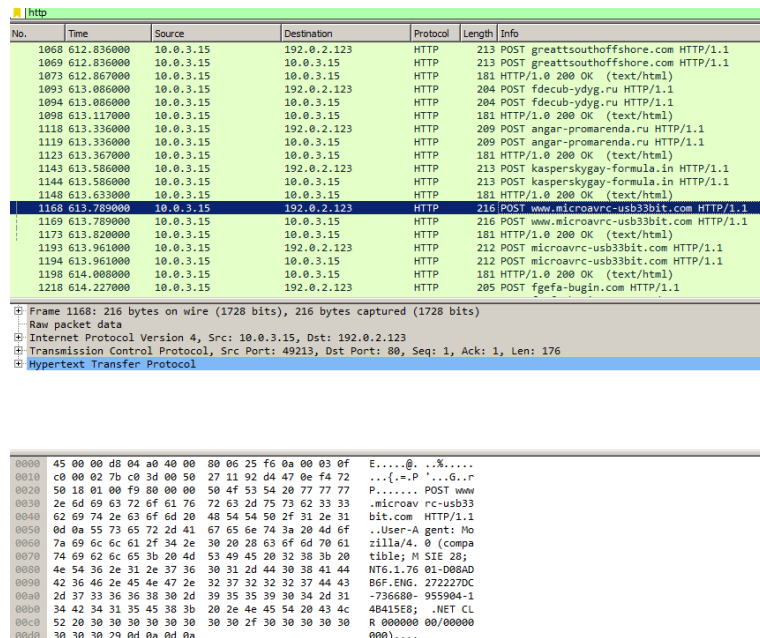
Avendo notato le connessioni HTTP mediante Fakenet, si è provato ad usare **Wireshark** per analizzare il traffico di rete individuando effettivamente delle richieste che puntavano a siti sospetti come ad esempio *www.microavr-usb33bit.com* visibile anche nell'immagine sottostante.



No.	Time	Source	Destination	Protocol	Length	Info
1041	590.820000	10.0.3.15	192.0.2.123	HTTP	235	GET /msdownload/update/v3/static/trusted/en/authro
1042	590.820000	10.0.3.15	10.0.3.15	HTTP	235	GET /msdownload/update/v3/static/trusted/en/authro
1046	590.820000	10.0.3.15	10.0.3.15	HTTP	181	HTTP/1.0 200 OK (text/html)
1068	612.836000	10.0.3.15	192.0.2.123	HTTP	213	POST greattsouthoffshore.com HTTP/1.1
1069	612.836000	10.0.3.15	10.0.3.15	HTTP	213	POST greattsouthoffshore.com HTTP/1.1
1073	612.867000	10.0.3.15	10.0.3.15	HTTP	181	HTTP/1.0 200 OK (text/html)
1093	613.086000	10.0.3.15	192.0.2.123	HTTP	204	POST fdecub-ydyg.ru HTTP/1.1
1094	613.086000	10.0.3.15	10.0.3.15	HTTP	204	POST fdecub-ydyg.ru HTTP/1.1
1098	613.117000	10.0.3.15	10.0.3.15	HTTP	181	HTTP/1.0 200 OK (text/html)
1118	613.336000	10.0.3.15	192.0.2.123	HTTP	209	POST angar-promarenda.ru HTTP/1.1
1119	613.336000	10.0.3.15	10.0.3.15	HTTP	209	POST angar-promarenda.ru HTTP/1.1
1123	613.367000	10.0.3.15	10.0.3.15	HTTP	181	HTTP/1.0 200 OK (text/html)
1143	613.586000	10.0.3.15	192.0.2.123	HTTP	213	POST kasperskygay-formula.in HTTP/1.1
1144	613.586000	10.0.3.15	10.0.3.15	HTTP	213	POST kasperskygay-formula.in HTTP/1.1
1148	613.633000	10.0.3.15	10.0.3.15	HTTP	181	HTTP/1.0 200 OK (text/html)
1168	613.789000	10.0.3.15	192.0.2.123	HTTP	216	POST www.microavrc-usb33bit.com HTTP/1.1
1169	613.789000	10.0.3.15	10.0.3.15	HTTP	216	POST www.microavrc-usb33bit.com HTTP/1.1
1173	613.820000	10.0.3.15	10.0.3.15	HTTP	181	HTTP/1.0 200 OK (text/html)
1193	613.961000	10.0.3.15	192.0.2.123	HTTP	212	POST microavrc-usb33bit.com HTTP/1.1
1194	613.961000	10.0.3.15	10.0.3.15	HTTP	212	POST microavrc-usb33bit.com HTTP/1.1
1198	614.008000	10.0.3.15	10.0.3.15	HTTP	181	HTTP/1.0 200 OK (text/html)
1218	614.227000	10.0.3.15	192.0.2.123	HTTP	205	POST fgefa-bugin.com HTTP/1.1
1219	614.227000	10.0.3.15	10.0.3.15	HTTP	205	POST fgefa-bugin.com HTTP/1.1

Figura 25: Wireshark - Image with mentioned site and others

Mostriamo qui un focus sulla richiesta di connessione HTTP del virus evidenziata da Wire-shark.



No.	Time	Source	Destination	Protocol	Length	Info
1068	612.836000	10.0.3.15	192.0.2.123	HTTP	213	POST greattsouthoffshore.com HTTP/1.1
1069	612.836000	10.0.3.15	10.0.3.15	HTTP	213	POST greattsouthoffshore.com HTTP/1.1
1073	612.867000	10.0.3.15	10.0.3.15	HTTP	181	HTTP/1.0 200 OK (text/html)
1093	613.086000	10.0.3.15	192.0.2.123	HTTP	204	POST fdecub-ydyg.ru HTTP/1.1
1094	613.086000	10.0.3.15	10.0.3.15	HTTP	204	POST fdecub-ydyg.ru HTTP/1.1
1098	613.117000	10.0.3.15	10.0.3.15	HTTP	181	HTTP/1.0 200 OK (text/html)
1118	613.336000	10.0.3.15	192.0.2.123	HTTP	209	POST angar-promarenda.ru HTTP/1.1
1119	613.336000	10.0.3.15	10.0.3.15	HTTP	209	POST angar-promarenda.ru HTTP/1.1
1123	613.367000	10.0.3.15	10.0.3.15	HTTP	181	HTTP/1.0 200 OK (text/html)
1143	613.586000	10.0.3.15	192.0.2.123	HTTP	213	POST kasperskygay-formula.in HTTP/1.1
1144	613.586000	10.0.3.15	10.0.3.15	HTTP	213	POST kasperskygay-formula.in HTTP/1.1
1148	613.633000	10.0.3.15	10.0.3.15	HTTP	181	HTTP/1.0 200 OK (text/html)
1168	613.789000	10.0.3.15	192.0.2.123	HTTP	216	POST www.microavrc-usb33bit.com HTTP/1.1
1169	613.789000	10.0.3.15	10.0.3.15	HTTP	216	POST www.microavrc-usb33bit.com HTTP/1.1
1173	613.820000	10.0.3.15	10.0.3.15	HTTP	181	HTTP/1.0 200 OK (text/html)
1193	613.961000	10.0.3.15	192.0.2.123	HTTP	212	POST microavrc-usb33bit.com HTTP/1.1
1194	613.961000	10.0.3.15	10.0.3.15	HTTP	212	POST microavrc-usb33bit.com HTTP/1.1
1198	614.008000	10.0.3.15	10.0.3.15	HTTP	181	HTTP/1.0 200 OK (text/html)
1218	614.227000	10.0.3.15	192.0.2.123	HTTP	205	POST fgefa-bugin.com HTTP/1.1

Frame 1168: 216 bytes on wire (1728 bits), 216 bytes captured (1728 bits)

Raw packet data

Internet Protocol Version 4, Src: 10.0.3.15, Dst: 192.0.2.123

Transmission Control Protocol, Src Port: 49213, Dst Port: 80, Seq: 1, Ack: 1, Len: 176

Hypertext Transfer Protocol

```
0000 45 00 00 d8 04 a0 00 00 80 06 25 f6 0a 00 03 0f E.....@. ..X....
0010 c0 00 02 7b c8 3d 00 50 27 11 92 44 47 0e f4 72 ...{..P'..G..r
0020 50 18 01 00 f9 80 00 00 50 4f 53 54 20 77 77 77 P..... POST ww
0030 2e 6d 69 63 72 6f 61 76 72 63 2d 75 73 62 33 33 .microav rc-usb33
0040 62 69 74 2e 63 6f 6d 20 48 54 54 50 2f 31 2e 31 bit.com HTTP/1.1
0050 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f ..User-A gent: Mo
0060 7a 69 6c 6c 61 2f 34 2e 30 20 28 63 6f 6d 70 61 zilla/4.0 (compa
0070 74 69 62 6c 65 3b 20 4d 53 49 45 20 32 3b 20 3b tible; M SIE 28;
0080 4e 54 36 2e 31 2e 37 36 30 31 2d 44 30 30 41 44 NT6.1.76 01-D08AD
0090 42 36 46 2e 45 4e 47 2e 32 37 32 32 37 44 43 B6F.ENG. 272227DC
00a0 2d 37 33 36 38 30 2d 39 35 35 39 30 34 2d 31 -736680- 955904-1
00b0 34 42 34 31 35 45 38 30 20 2e 4e 45 54 20 43 4c 4B415EB; .NET CL
00c0 52 20 30 30 30 30 30 30 30 30 2f 30 30 30 30 R 000000 00/00000
00d0 30 30 30 20 0d 0a 0d 0a 000)....
```

Figura 26: Wireshark - *www.microavr-usb33bit.com*

## Parte III

# Reverse Engineering

Dai comportamenti emersi in analisi dinamica, si osserva che l'eseguibile non è una semplice calcolatrice.

Il Reverse Engineering è una tecnica per decomporre un oggetto, capirne il funzionamento, analizzandone a fondo il codice macchina che crea il sorgente del linguaggio originale. Utilizzando queste tecniche di Reverse Engineering con tools **IDA** e **x32dbg** l'obiettivo è quello di individuare zone di codice in cui si effettuano operazioni sospette. Queste operazioni sospette sono:

- Disabilitazione Security Center
- Modifica dei registri
- Invio di pacchetti in rete

Per scoprire tutte queste azioni occorre deoffuscare il codice nella sezione `.vmp0` usando il tool `x32dbg`.

Con l'uso del tool `x32dbg` si notano delle chiamate di sistema che lanciano il prompt dei comandi il quale eseguirà delle operazioni da riga di comando. Queste operazioni risultano piuttosto sospette per una calcolatrice di Windows.

Nella figura sotto vediamo il punto esatto della chiamata a `SHELL32.dll`.

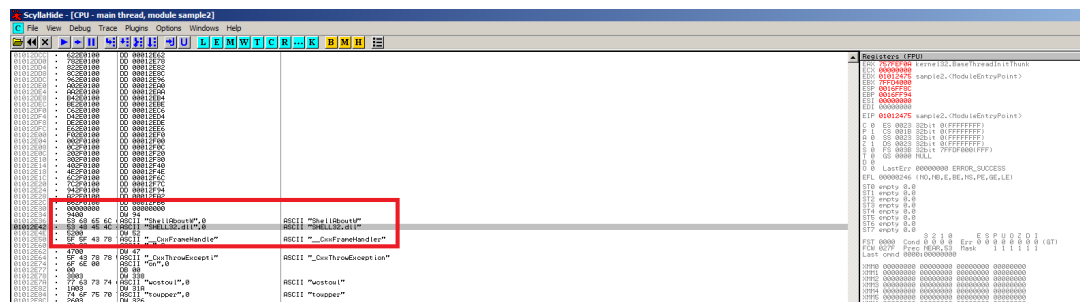


Figura 27: `x32dbg` - shell system call

## Offuscamento

Il codice presenta due sezioni di interesse: quella in chiaro `.text` e quella offuscata `.vmp0`. La sezione `.vmp0` non è solamente offuscata ma sembrerebbe essere criptata, risulta quindi illeggibile fino a quando non viene eseguito il codice all'interno della sezione `.text`, che decripta la sezione `.vmp0`. Di conseguenza, la sezione `.text` dovrà contenere la chiave di lettura. La sezione in `.text`, nonostante non sia criptata, risulta comunque molto complessa da analizzare quindi anche in questa sezione risulta essere offuscata.

Ora procederemo ad analizzare il codice relativo alla disattivazione di Windows Security Center con `x32dbg` e successivamente analizzeremo il codice relativo al salvataggio dei file, ovvero di `usr28zt32.dll`.

L'eseguibile, come detto dall'analisi con `PEID`, non è packed.

```

C:\Administrator: C:\Windows\system32\cmd.exe
05/12/2016 17:11 <DIR> Python27
22/05/2019 19:55 <DIR> Tools
05/12/2016 11:04 <DIR> Users
22/05/2019 19:48 <DIR> Windows
                2 File(s)      34 bytes
                9 Dir(s)  10,703,093,760 bytes free

C:\>cd Tools
C:\Tools>cd upx
C:\Tools\upx>upx -d -o unpacke.exe C:\Users\Malware\Desktop\Samples\Other\sample
2.exe
                Ultimate Packer for eXecutables
                Copyright (C) 1996 - 2013
UPX 3.91w      Markus Oberhumer, Laszlo Molnar & John Reiser   Sep 30th 2013

-----
File size      Ratio      Format      Name
-----
upx: C:\Users\Malware\Desktop\Samples\Other\sample2.exe: NotPackedException: not
packed by UPX

Unpacked 0 files.
C:\Tools\upx>

```

Figura 28: *cmd - Error upx*

Upx quindi non è stato usato per comprimere il virus.

## Disattivazione Security Center

La disattivazione del Security Center su Windows 7, a differenza di Windows 10, non richiede i privilegi di amministratore. Nell'elenco dei servizi presenti nello stack non è infatti presente quello relativo al Security Center.

Il virus quando esegue *dllhost.exe*, durante l'infezione, viene intrapreso un ramo di esecuzione differente all'interno del quale viene effettuata la chiamata alla funzione "StartServiceW", all'interno della quale viene passato il parametro per la libreria "ComSysApp" (figura 2 - riga evidenziata), la quale si occupa della gestione delle dll (librerie) volte alla disattivazione del firewall di Windows.

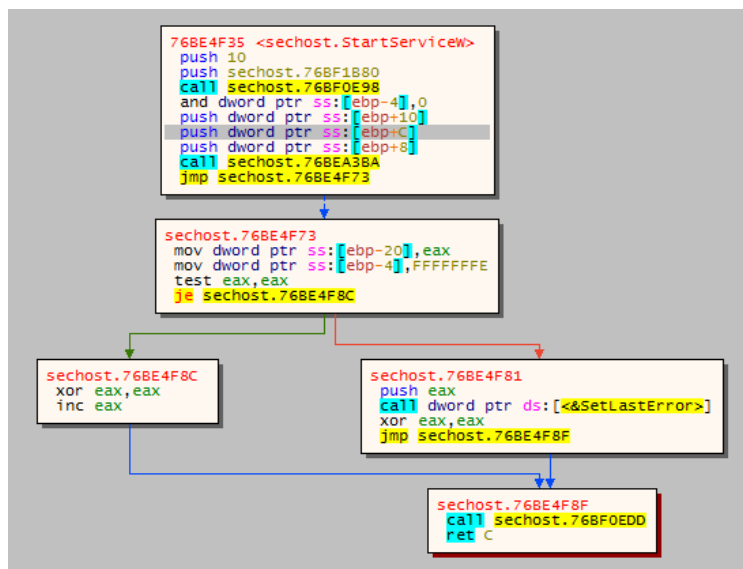


Figura 29: *x32dbg - Call to "StartService"*

## Salvataggio dei file

Il virus è di tipo *Expiro* per cui si è ricercato qualche tipo di file che servisse per salvare i dati nella libreria della forma `wsr**zt32.dll`. Il riscontro è avvenuto con il file `wsr28zt32.dll` già trovato nella fase di analisi dinamica. Di seguito si mostra l'immagine che evidenzia il salvataggio di file nella libreria sopra citata.

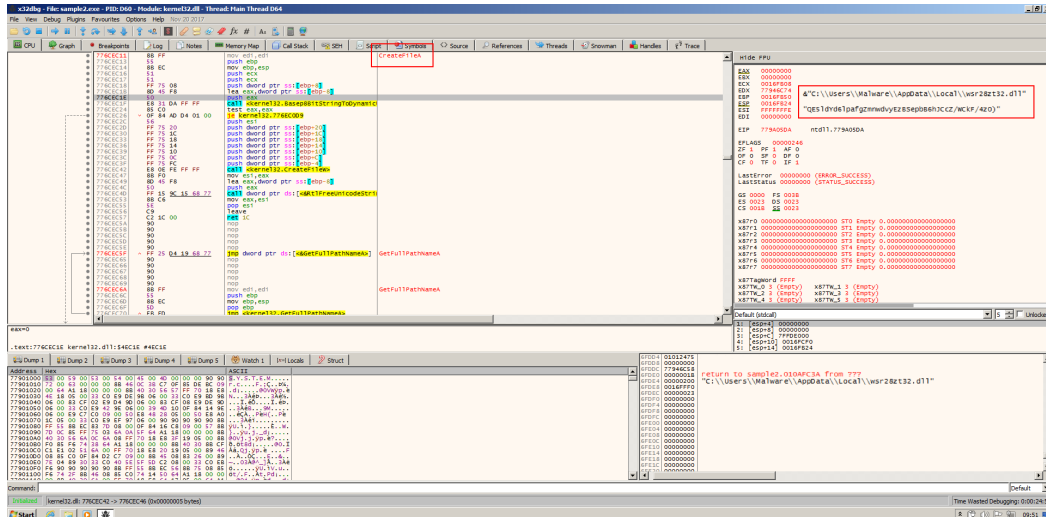


Figura 30: File creation

```
QE51dyd61pafgzmnwdvyeZBsepBB6hJCCZ/WckF/4z0Ww47PE1qnICKaqj(ZE8rph1j  
swtqqv7nn9q/10Mndcjva45jva4(YQL1wtvuoJUMuvMRh7o2vTPfgXij1SU2b23JNMOTXUE  
QfARWAGCEfRA5JHZAv2ThkarETwCERM6QHeqXQLSkSxDChDEatYQ4EMckY0Dy5KmQUUQKAM7  
xFa/q5Si5Ixd1wwx21pod1PRjmmZahQ731MJWypoyykvj4vYox4f1fMuvDYZIquvFIWakBf
```

Figura 31: Contents of the file



La figura sotto mostra la procedura di come il virus riesce a creare i file.

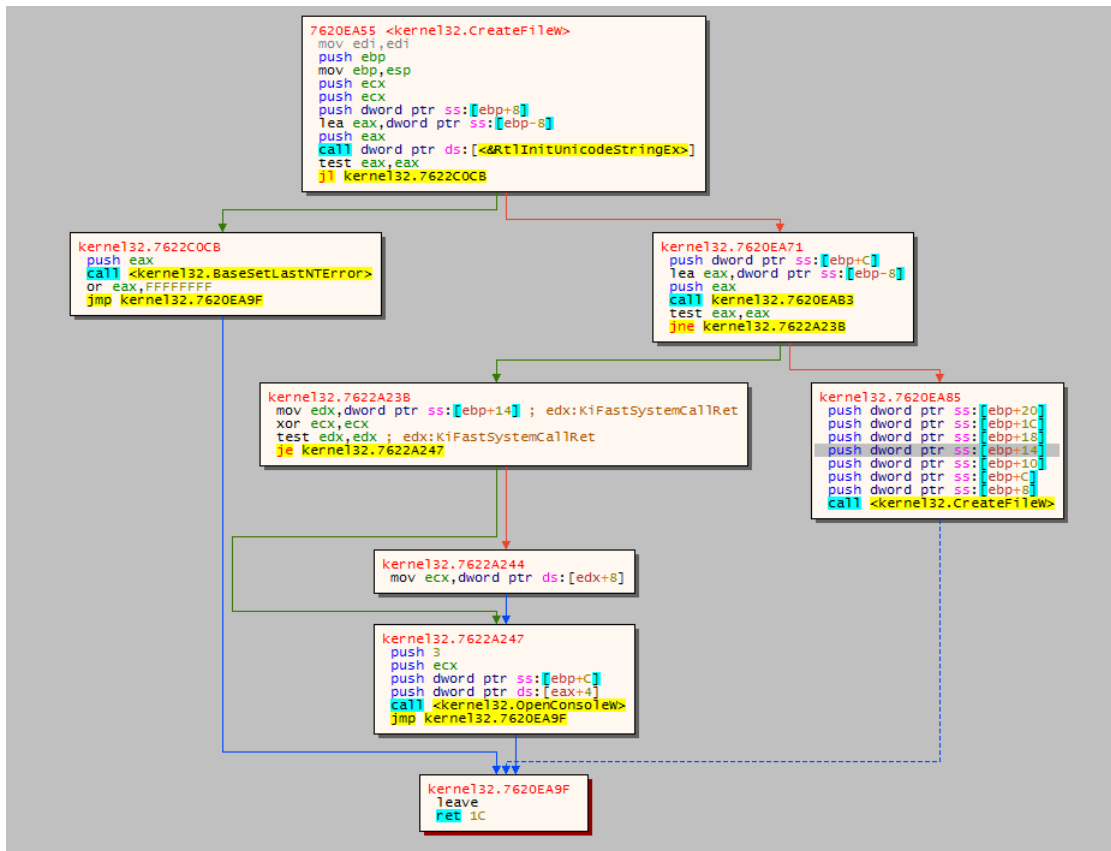


Figura 32: Subroutine that creates any file

## Parte IV

# Conclusioni

Di seguito i report di analisi statica e dinamica.

Quella statica risulta essere molto meno precisa rispetto a quella dinamica. L'*analisi statica* mette in evidenza la presenza di sezioni malevole e offuscate così come la mancanza di firma digitale e di certificato. L'*analisi dinamica* profila esattamente il comportamento di virus Expiro che cerca di prelevare dati dalla macchina ed inviarli successivamente in rete.

Static Analysis		
Category	Select	Score
Packed		0
Strings		3
Imports		2
Sections		1
Main Icon		1
Additional Icons		0
Dialogs		0
Version Information		0
Digital Signature		2
Total Score		9
Verdict		Potentially Suspicious

Dynamic Analysis		
Category	Select	Score
Persistence		2
File Manipulation		2
Process Manipulation		2
Registry Manipulation		2
Additional Processes		0
Removal Resistance		2
Analysis Resistance		2
Interface/Visible Activity		0
Network Activity		2
Rootkit Behaviour		0
System Calls		1
Behaviour		2
Total Score		17
Verdict		Suspicious

Figura 33: Static and Dynamic Analysis

Come già detto, il malware analizzato si maschera da normale calcolatrice di Windows usando la medesima icona. In background però commette diverse azioni malevole tra cui: accedere a determinate chiavi nel registro salvando il loro valore nei file opportuni, modifica degli eseguibili (in particolare i browser con capacità di indirizzamento automatico ad un elenco di siti malevoli) ed infine disattivazione dei sistemi di protezione locali tra i quali Windows Defender.

Le informazioni vengono inviate all'esterno attraverso richieste HTTP.

Dall'analisi mediante i tool *IDA* e *x32dbg*, si notano tecniche di offuscamento del codice e ciò è evidente dall'analisi della sezione *.vmp0*.

## Bibliografia

<https://www.microsoft.com/en-us/wdsi/threats/malware-encyclopedia-description?Name=Win32/Expiro>

<https://www.virustotal.com/gui/file/34558ac3bfab17ca1a1ff70860b35296395f1df7fa8d86b39c56faecf9c3cffc>

[https://en.wikipedia.org/wiki/Microsoft\\_Windows\\_library\\_files](https://en.wikipedia.org/wiki/Microsoft_Windows_library_files)