

Manuel Di Gangi

S10_L2

Analisi dinamica basica

26 marzo 2024

INDICE

Traccia.....	2
1. Librerie importate.....	3
2. Sezioni.....	4
3. Considerazioni finali.....	4

Traccia

Configurare la macchina virtuale per l'analisi dinamica (il malware sarà effettivamente eseguito). Con riferimento al file eseguibile contenuto nella cartella «Esercizio_Pratico_U3_W2_L2» presente sul desktop della vostra macchina virtuale dedicata all'analisi dei malware, rispondere ai seguenti quesiti:

- Identificare eventuali azioni del malware sul file system utilizzando Process Monitor (procmon)
- Identificare eventuali azioni del malware sui processi e thread utilizzando Process Monitor
- Modifiche del registro dopo il malware(le differenze)
- Provare a profilare il malware in base alla correlazione tra «operation» e Path.

1. Analisi statica basica

1.1 Librerie importate

Con l'ausilio del tool CFF Explorer troviamo che il malware richiama le seguenti librerie:

- **KERNEL32.dll**: contiene le funzioni principali per interagire con il sistema operativo, ad esempio: manipolazione dei file, la gestione della memoria.

Osservando la tabella a fianco possiamo osservare che l'unica libreria richiamata (KERNEL32) importa ben 54 funzioni tra le quali troviamo:

- **CloseHandle**: è un'API utilizzata nei sistemi operativi Windows per chiudere un handle (un tipo di riferimento o puntatore) a un oggetto kernel. In sostanza, questa funzione viene utilizzata per rilasciare le risorse utilizzate da un handle dopo che è stato utilizzato.
- **GetProcAddress** è una funzione utilizzata principalmente nella programmazione di Windows, ha la funzione di libreria dinamica che consente ai programmatori di ottenere un puntatore a una funzione all'interno di una DLL caricata in memoria. Questo è spesso utilizzato quando si desidera chiamare una funzione esportata da una DLL in modo dinamico durante l'esecuzione del programma, anziché linkarla staticamente durante la compilazione.

Module Name	Imports	OFTs	TimeDateStamp	ForwarderChai
000046B8	N/A	00004444	00004448	0000444C
szAnsi	(nFunctions)	Dword	Dword	Dword
KERNEL32.dll	54	0000446C	00000000	00000000
OFTs	FTs (IAT)	Hint	Name	
Dword	Dword	Word	szAnsi	
00004548	00004548	001B	CloseHandle	
00004556	00004556	02BF	VirtualFree	
00004564	00004564	0218	ReadFile	
00004570	00004570	02BB	VirtualAlloc	
00004580	00004580	0112	GetFileSize	
0000458E	0000458E	0034	CreateFileA	

- **VirtualProtect** è una funzione presente nel linguaggio di programmazione Microsoft Windows, viene utilizzata per modificare i permessi di accesso alla memoria di un processo in esecuzione. Tipicamente, viene utilizzata per cambiare i permessi di una pagina di memoria da scrivibile a solo-lettura o eseguibile, o viceversa. Questo può essere utile per scopi di sicurezza, come proteggere parti critiche della memoria da modifiche non autorizzate o per consentire l'esecuzione di codice generato dinamicamente.
- **VirtualAlloc** è una funzione di programmazione utilizzata principalmente nella piattaforma Windows per riservare o allocare memoria virtuale per un processo. Essa consente ai programmatori di riservare una porzione di memoria virtuale senza necessariamente allocare memoria fisica corrispondente nello spazio di archiviazione fisica (RAM o disco rigido).
- **VirtualFree** è una funzione di Windows API utilizzata per liberare la memoria allocata dinamicamente da un processo.

1.2 Sezioni

Dal menù “section header” vediamo che il malware è composto da 4 sezioni

.text	00002E96	00001000	00003000	00001000	00000000	00000000	0000	0000	60000020
.rdata	000008F2	00004000	00001000	00004000	00000000	00000000	0000	0000	40000040
.data	000007DC	00005000	00001000	00005000	00000000	00000000	0000	0000	C0000040
.rsrc	00006084	00006000	00007000	00006000	00000000	00000000	0000	0000	40000040

- **.text**: contiene le istruzioni (le righe di codice) che la CPU eseguirà una volta che il software sarà avviato. Generalmente questa è l'unica sezione di un file eseguibile che viene eseguita dalla CPU, in quanto tutte le altre sezioni contengono dati o informazioni a supporto.
- **.rdata**: include generalmente le informazioni circa le librerie e le funzioni importate ed esportate dall'eseguibile, informazione che come abbiamo visto possiamo ricavare con CFF Explorer.
- **.data**: contiene tipicamente i dati / le variabili globali del programma eseguibile, che devono essere disponibili da qualsiasi parte del programma. Una variabile si dice globale quando non è definita all'interno di un contesto di una funzione, ma bensì è globalmente dichiarata ed è di conseguenza accessibile da qualsiasi funzione all'interno dell'eseguibile.
- **.rsrc**: include le risorse utilizzate dall'eseguibile come ad esempio icone, immagini, menu e stringhe che non sono parte dell'eseguibile stesso.

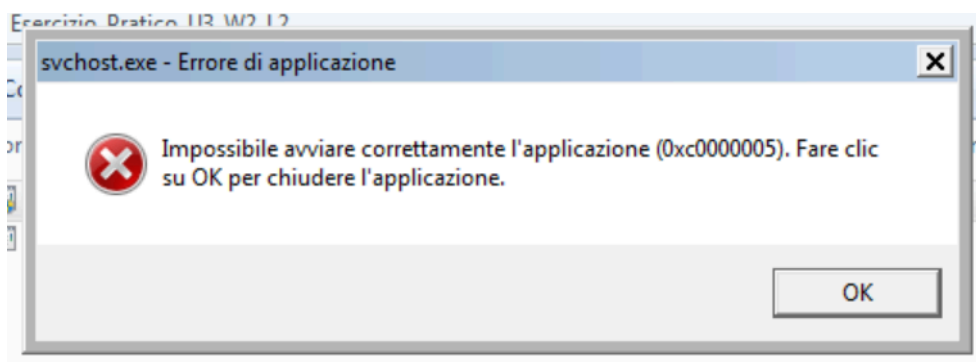
2. Analisi dinamica basica

Prima di lanciare il malware avviamo tutti gli strumenti di cui necessitiamo per effettuare lo studio del suo comportamento. Dall'analisi statica possiamo intuire che questo malware non effettua connessioni quindi non necessitiamo degli strumenti come Wireshark.

Per lo studio del comportamento del programma utilizzeremo:

- **Procmon:** un tool avanzato per Windows che permette di monitorare i processi ed i thread attivi, l'attività di rete, l'accesso ai file e le chiamate di sistema effettuate su un sistema operativo.
- **Reghost:** un tool che permette di paragonare due istantanee delle chiavi di registro salvate in due momenti separati tra di loro.
- **ProcessExplorer:** ci da una vista sui processi in esecuzione.

All'avvio del programma osserviamo un comportamento anomalo, anche per un malware. Il programma parte ma viene interrotto immediatamente restituendo un messaggio di errore. Dallo studio presentato nelle pagine seguenti vedremo che questo malware va ad infettare il processo di sistema **svchost.exe**, ma essendo studiato per windows XP su windows 7 non funziona come dovrebbe. Per proseguire lo studio abbiamo spostato il malware ed i tool sulla VM di Windows XP.



Con l'ausilio di ProcMon andiamo a studiare l'esecuzione del programma

Avvio del malware

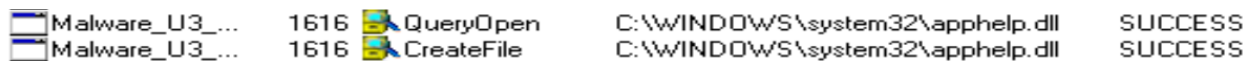


Subito dopo l'avvio il malware esegue l'operazione **"Create File"** la quale crea un file se il file non esiste, nel caso sia già presente, apre il file e lo modifica. Già da questa operazione ci rendiamo conto che il malware va ad infettare un processo vitale per il sistema.

Svchost.exe è un processo di sistema che può ospitare uno o più servizi Windows nella famiglia di sistemi operativi Windows NT. Svchost è essenziale nell'implementazione di processi di servizi condivisi, in cui più servizi possono condividere un processo per ridurre il consumo di risorse.



Come per svchost il malware continua a modificare altre librerie come quella nell'immagine seguente, eseguendo prima la query di apertura del file per verificare la correttezza del path per poi andare a modificarne il contenuto.




Proseguendo, osserviamo che il malware effettua una specie di Brute force del path, prova tutti i percorsi del sistema fino a quando non trova il processo, libreria o come nell'immagine seguente la chiave che sta cercando.

[illegible]


Man mano che le informazioni vengono ottenute il malware va di volta in volta a modificare il file svchost.exe, probabilmente andando a modificare le info ottenute in precedenza.

Malware_U3...	1616	RegOpenKey	HKCU	SUCCESS
Malware_U3...	1616	RegOpenKey	HKCU\Software\Policies\Microsoft\Con...	NAME NOT FOUND
Malware_U3...	1616	RegOpenKey	HKCU\Control Panel\Desktop	SUCCESS
Malware_U3...	1616	RegQueryValue	HKCU\Control Panel\Desktop\MultiUIL...	NAME NOT FOUND
Malware_U3...	1616	RegCloseKey	HKCU\Control Panel\Desktop	SUCCESS
Malware_U3...	1616	RegCloseKey	HKCU	SUCCESS
Malware_U3...	1616	QueryOpen	C:\WINDOWS\system32\svchost.exe	SUCCESS
Malware_U3...	1616	CreateFile	C:\WINDOWS\system32\svchost.exe	SUCCESS
Malware_U3...	1616	CreateFileMap...	C:\WINDOWS\system32\svchost.exe	SUCCESS
Malware_U3...	1616	QueryStandard...	C:\WINDOWS\system32\svchost.exe	SUCCESS
Malware_U3...	1616	CreateFileMap...	C:\WINDOWS\system32\svchost.exe	SUCCESS
Malware_U3...	1616	CloseFile	C:\WINDOWS\system32\svchost.exe	SUCCESS
Malware_U3...	1616	QueryOpen	C:\WINDOWS\system32\svchost.exe	SUCCESS
Malware_U3...	1616	CreateFile	C:\WINDOWS\system32\svchost.exe	SUCCESS
Malware_U3...	1616	CreateFileMap...	C:\WINDOWS\system32\svchost.exe	SUCCESS
Malware_U3...	1616	QueryStandard...	C:\WINDOWS\system32\svchost.exe	SUCCESS
Malware_U3...	1616	CreateFileMap...	C:\WINDOWS\system32\svchost.exe	SUCCESS
Malware_U3...	1616	CloseFile	C:\WINDOWS\system32\svchost.exe	SUCCESS

terminata questa fase, il malware crea un file all'interno della stessa cartella da cui è stato avviato. Come possiamo vedere dall'immagine sottostante, andando a filtrare la vista di procmon sul file in oggetto, vengono eseguite delle operazioni di "append" e lo deduciamo dal fatto che tale file aumenta di lunghezza nel tempo.



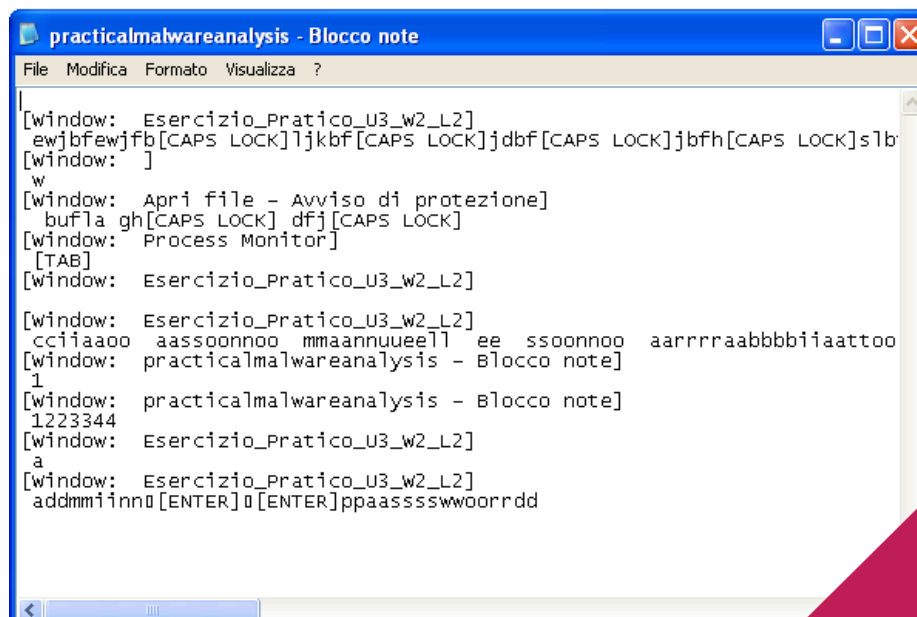
Malware_U3_W2_L2



practicalmalwareanalysis
Documento di testo
1 KB

svchost...	348	CreateFile	C:\Documents and Settings\All Users\Documenti\Esercizio_...	SUCCESS	Desired Access: Generic Wri...
svchost...	348	QueryStandardI...	C:\Documents and Settings\All Users\Documenti\Esercizio_...	SUCCESS	AllocationSize: 0, EndOfFile: ...
svchost...	348	WriteFile	C:\Documents and Settings\All Users\Documenti\Esercizio_...	SUCCESS	Offset: 0, Length: 12
svchost...	348	WriteFile	C:\Documents and Settings\All Users\Documenti\Esercizio_...	SUCCESS	Offset: 12, Length: 26
svchost...	348	WriteFile	C:\Documents and Settings\All Users\Documenti\Esercizio_...	SUCCESS	Offset: 38, Length: 4
svchost...	348	WriteFile	C:\Documents and Settings\All Users\Documenti\Esercizio_...	SUCCESS	Offset: 42, Length: 1
svchost...	348	CloseFile	C:\Documents and Settings\All Users\Documenti\Esercizio_...	SUCCESS	
svchost...	348	CreateFile	C:\Documents and Settings\All Users\Documenti\Esercizio_...	SUCCESS	Desired Access: Generic Wri...
svchost...	348	QueryStandardI...	C:\Documents and Settings\All Users\Documenti\Esercizio_...	SUCCESS	AllocationSize: 48, EndOfFile: ...
svchost...	348	WriteFile	C:\Documents and Settings\All Users\Documenti\Esercizio_...	SUCCESS	Offset: 43, Length: 1
svchost...	348	CloseFile	C:\Documents and Settings\All Users\Documenti\Esercizio_...	SUCCESS	
svchost...	348	CreateFile	C:\Documents and Settings\All Users\Documenti\Esercizio_...	SUCCESS	Desired Access: Generic Wri...
svchost...	348	QueryStandardI...	C:\Documents and Settings\All Users\Documenti\Esercizio_...	SUCCESS	AllocationSize: 48, EndOfFile: ...
svchost...	348	WriteFile	C:\Documents and Settings\All Users\Documenti\Esercizio_...	SUCCESS	Offset: 44, Length: 1
svchost...	348	CloseFile	C:\Documents and Settings\All Users\Documenti\Esercizio_...	SUCCESS	
svchost...	348	CreateFile	C:\Documents and Settings\All Users\Documenti\Esercizio_...	SUCCESS	Desired Access: Generic Wri...
svchost...	348	QueryStandardI...	C:\Documents and Settings\All Users\Documenti\Esercizio_...	SUCCESS	AllocationSize: 48, EndOfFile: ...
svchost...	348	WriteFile	C:\Documents and Settings\All Users\Documenti\Esercizio_...	SUCCESS	Offset: 45, Length: 1
svchost...	348	CloseFile	C:\Documents and Settings\All Users\Documenti\Esercizio_...	SUCCESS	
svchost...	348	CreateFile	C:\Documents and Settings\All Users\Documenti\Esercizio_...	SUCCESS	Desired Access: Generic Wri...
svchost...	348	QueryStandardI...	C:\Documents and Settings\All Users\Documenti\Esercizio_...	SUCCESS	AllocationSize: 48, EndOfFile: ...
svchost...	348	WriteFile	C:\Documents and Settings\All Users\Documenti\Esercizio_...	SUCCESS	Offset: 46, Length: 1

recandoci nella directory e aprendo tale documento osserviamo che sono stati registrati tutti i tasti premuti dall'utente



```

practicalmalwareanalysis - Blocco note
File Modifica Formato Visualizza ?

[window: Esercizio_Pratico_U3_W2_L2]
ewjbfewjfb[CAPS LOCK]ljkbf[CAPS LOCK]jdbf[CAPS LOCK]jbfh[CAPS LOCK]slb
[window: ]
w
[window: Apri file - Avviso di protezione]
buf1a gh[CAPS LOCK] dfj[CAPS LOCK]
[window: Process Monitor]
[TAB]
[window: Esercizio_Pratico_U3_W2_L2]

[window: Esercizio_Pratico_U3_W2_L2]
cc1iaaoo aassoonnoo mmaannuueell ee ssoonnoo aarrrraabbbbiiaattoo
[window: practicalmalwareanalysis - Blocco note]
1
[window: practicalmalwareanalysis - Blocco note]
1223344
[window: Esercizio_Pratico_U3_W2_L2]
a
[window: Esercizio_Pratico_U3_W2_L2]
addmmiinno[ENTER][ENTER]ppaasssswwoorrdd

```

3. Conclusioni

Al termine dell'analisi possiamo concludere che il malware studiato sia un Keylogger, che registra i tasti digitati dall'utente ignaro e li salva in un documento sulla macchina locale, in attesa che un utente malevolo lo vada a recuperare in un secondo momento. Questo genere di malware è molto semplice quanto pericoloso dal momento che potrebbe registrare anche i dati di accesso dell'utente, il che lo rende una minaccia non indifferente e difficile da estirpare dal momento che come visto in precedenza è andato a modificare molti file di sistema.