

FUNZIONALITA' DEI MALWARE

M6 - MALWARE ANALYSIS



Prerequisiti

Tra le buone pratiche per configurare un ambiente sicuro:

Configurazione schede di rete: non deve avere accesso diretto ad Internet e preferibilmente nemmeno ad altre macchine sulla rete. La configurazione ideale è

I) Eliminare le interfacce di rete durante l'analisi statica

II) Abilitare un'interfaccia di rete interna (su VirtualBox viene chiamata 'rete interna' per l'analisi dinamica).

Questa impostazione è necessaria per monitorare il traffico che genera potenzialmente il malware

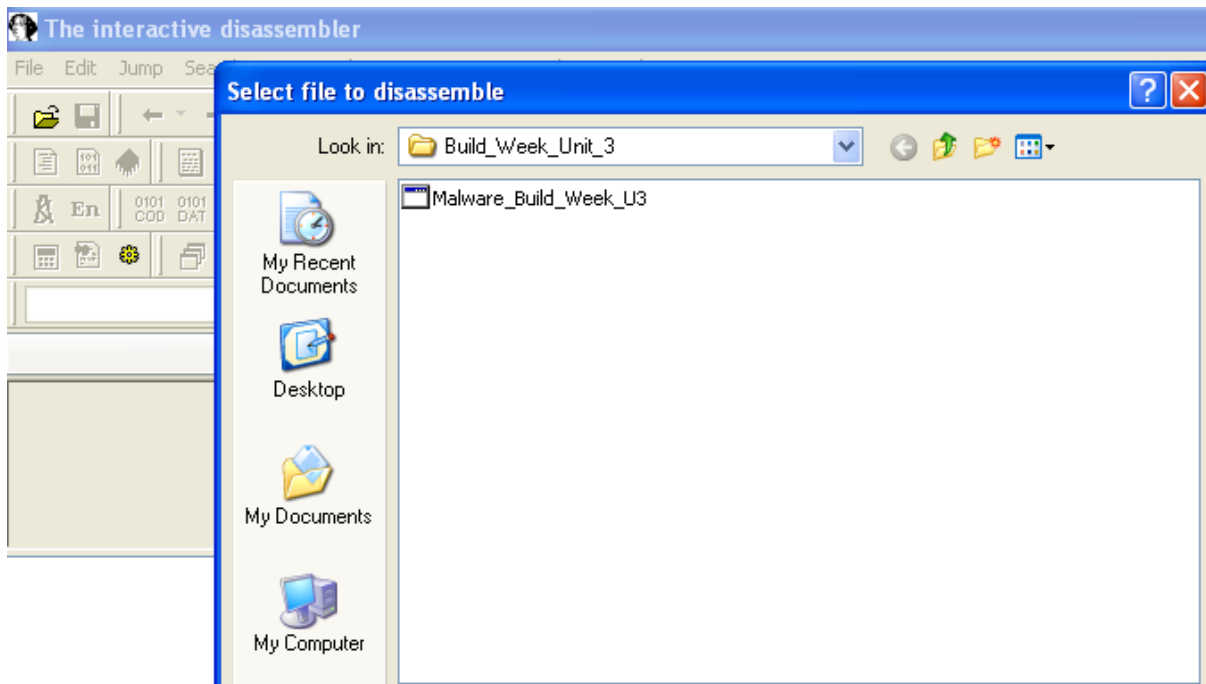
Dispositivi USB: quando un dispositivo USB viene collegato alla macchina fisica, esso può essere riconosciuto anche dall'ambiente di test. Per evitare ciò, è buona pratica non abilitare o disabilitare il controller USB. Il malware infatti potrebbe utilizzare il dispositivo USB per propagarsi sulla macchina fisica

Cartelle condivise: le cartelle condivise tra la nostra macchina e il laboratorio virtuale possono essere utilizzate dal malware per propagarsi al di fuori del laboratorio causando danni alla nostra macchina e alla rete domestica. Non dobbiamo quindi condividere cartelle tra host e guest.

Creare delle istantanee: possiamo creare istantanee della macchina virtuale nel suo stato iniziale per avere la possibilità di ripristinarlo qualora ce ne fosse il bisogno.

Traccia

Il malware da analizzare è nella cartella Build_Week_Unit_3 presente sul desktop della macchina virtuale dedicata



ANALISI STATICA

Con riferimento al file eseguibile Malware_Build_Week_U3, rispondere ai seguenti quesiti utilizzando i tool e le tecniche apprese nelle lezioni teoriche:

Per l'analisi statica avanzata possiamo utilizzare Ida Pro, che risulta essere l'unico disassembler sul mercato..tuttavia riesce a mettere a disposizione una serie di caratteristiche intuitive per semplificare le attività. Infatti, oltre alla traduzione completa del linguaggio macchina di un eseguibile in linguaggio assembly, IDA identifica:

- funzioni/chiamate di funzione
- analisi dello stack
- variabili locali e parametri.

QUANTI PARAMETRI SONO PASSATI ALLA FUNZIONE MAIN()?

I parametri passati allaq funzione Main() sono:

- 1) argc
- 2) argv
- 3) envp

```

; Attributes: bp-based frame

; int __cdecl main(int argc, const char **argv, const char *envp)
_main proc near

hModule= dword ptr -11Ch
Data= byte ptr -118h
var_8= dword ptr -8
var_4= dword ptr -4
argc= dword ptr 8
argv= dword ptr 0Ch
envp= dword ptr 10h

push    ebp

```

QUANTE VARIABILI SONO DICHIARATE ALL'INTERNO DELLA FUNZIONE MAIN()?

Le variabili dichiarate all'interno della funzione Main() sono 4:

1. hModule
2. Data
3. var_8
4. var_4

```

; Attributes: bp-based frame

; int __cdecl main(int argc, const char **argv, const char *envp)
_main proc near

hModule= dword ptr -11Ch
Data= byte ptr -118h
var_8= dword ptr -8
var_4= dword ptr -4
argc= dword ptr 8
argv= dword ptr 0Ch
envp= dword ptr 10h

push    ebp

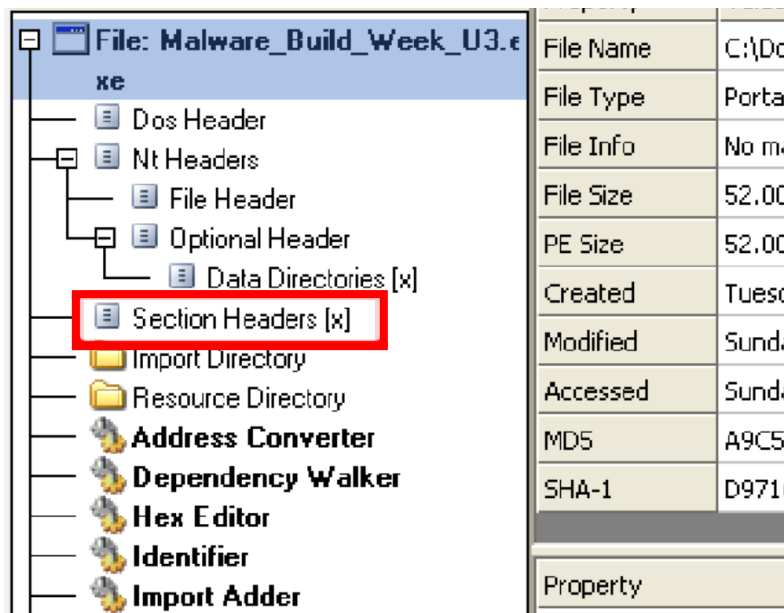
```

QUALI SEZIONI SONO PRESENTI ALL'INTERNO DEL FILE ESEGUIBILE?

Per poter analizzare le sezioni all'interno del file eseguibile, possiamo affidarci ad un altro tool: **CFF Explorer**: un software che ci permette di analizzare la struttura interna di file eseguibili. Tra le varie funzionalità troviamo:

- l'esplorazione delle sezioni dei file
- gestione delle risorse
- visualizzazioni delle intestazioni PE

Dall'interfaccia principale selezioniamo "Section Headers [x]" per poter visualizzare le sezioni del file eseguibile:



DESCRIVETE BREVEMENTE ALMENO 2 DI QUELLE IDENTIFICATE:

1. .text

Le istruzioni in linguaggio assembly all'interno di questa sezione, vengono eseguite sequenzialmente appena il programma viene avviato. La sezione rappresenta la parte principale del codice macchina che guida l'esecuzione del programma.

This section contains:

Code Entry Point: 00001487

CFF Explorer rettifica che l'entry point dell'eseguibile la possiamo trovare all'indirizzo 00001487

2. .rdata

Contiene dati di sola lettura: dati che il programma può leggere ma non può modificare durante l'esecuzione

This section contains:

Data: 00007000

Import Directory: 000074EC

Import Address Table Directory: 00007000

Questa sezione indica:

- la sezione .rdata inizia all'indirizzo 00007000

- l'indirizzo di memoria 'Import Directory' è 000074EC

- l'indirizzo 'Import Address Table Directory' (tabella indicante gli indirizzi delle funzioni specifiche all'interno delle librerie dinamiche è 00007000, cioè lo stesso indirizzo in cui inizia la sezione

3. .data

Questa sezione invece contiene dati variabili che il programma può leggere e modificare durante l'esecuzione

QUALI LIBRERIE IMPORTA IL MALWARE?

Il malware importa le librerie ADVAPI32 e KERNEL32

Module Name	Imports	OFTs	TimeStamp	ForwarderChain	Name RVA	FTs (IAT)
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	51	00007534	00000000	00000000	0000769E	0000700C
ADVAPI32.dll	2	00007528	00000000	00000000	000076D0	00007000



Address	Ordinal	Name	Library
00407000		RegSetValueExA	ADVAPI32
00407004		RegCreateKeyExA	ADVAPI32
004070C0		ReadFile	KERNEL32
004070C4		MultiByteToWideChar	KERNEL32
00407010		LockResource	KERNEL32
00407014		LoadResource	KERNEL32
004070B8		LoadLibraryA	KERNEL32
004070...		LCMapStringW	KERNEL32
004070C8		LCMapStringA	KERNEL32
00407094		HeapReAlloc	KERNEL32
0040703C		HeapFree	KERNEL32
00407080		HeapDestroy	KERNEL32
00407084		HeapCreate	KERNEL32
00407090		HeapAlloc	KERNEL32

PER OGNUNA DELLE LIBRERIE IMPORTATE, FATE DELLE IPOTESI SULLA BASE DELLA SOLA ANALISI STATICA DELLE FUNZIONALITA' CHE IL MALWARE POTREBBE IMPLEMENTARE. UTILIZZATE LE FUNZIONI CHE SONO RICHIAMATE ALL'INTERNO DELLE LIBRERIE PER SUPPORTARE LE VOSTRE IPOTESI.

➤ ADVAPI32

Iniziamo da ADVAPI 32 che contiene principalmente delle funzioni riguardanti servizi avanzati di API: contiene per esempio funzioni per

- modificare il Registro di sistema
- autenticazione
- autorizzazione
- crittografia
- gestione di eventi di sicurezza
- nonché la creazione e gestione dei servizi di Windows.

 00407000	RegSetValueExA	ADVAPI32
 00407004	RegCreateKeyExA	ADVAPI32

IPOTESI:

Il malware potrebbe utilizzare le funzioni sopra elencate per creare o modificare le chiavi di Registro infatti le funzioni:

RegSetKeyExA = viene utilizzata per creare una nuova chiave di registro o aprirne una già esistente

RegSetValueExA = consente l'aggiunta di un nuovo valore al Registro di sistema




➤ KERNEL32

Per quanto riguarda invece KERNEL32 è una delle librerie di sistema principali in Windows. Contiene funzioni di basso livello che sono necessari per il funzionamento delle applicazioni, inclusi:

- processi
- memoria
- gestione di file
- tempo
-

IPOTESI:

Queste funzioni possono contenere stringhe crittografate o dati che il malware potrebbe sfruttare in fase di esecuzione. Manipolare le risorse infatti, potrebbe sia nascondere il codice malevolo sia implementare delle funzionalità specifiche all'interno del programma infetto

 00407028	FindResourceA	KERNEL32
 00407038	ExitProcess	KERNEL32
 004070A4	CreateFileA	KERNEL32

Sulla base di queste informazioni potremmo dedurre che si tratti di un 'Dropper': un malware che contiene al suo interno un altro malware. Con le funzioni 'WriteFile' e 'CreateFile' infatti, il dropper potrebbe oltretutto salvarsi sul disco.

MALWARE ANALYSIS

Spiegare:

1. Lo scopo della funzione chiamata alla locazione di memoria 00401021

Lo scopo della funzione `RegCreateKeyExA` è quello di creare una determinata chiave del registro di sistema o, se già esistente, di aprirla.

```
.text:00401015      push     0                      ; Reserved
.text:00401017      push     offset SubKey         ; "SOFTWARE\\Microsoft\\Windows NT\\CurrentVe"...
.text:0040101C      push     00000000             ; hKey
.text:00401021      call     ds:RegCreateKeyExA
.text:00401027      test     eax, eax
.text:00401029      jz       short loc_401032
```

2. Come vengono passati i parametri alla funzione alla locazione 00401021

```
LSTATUS RegCreateKeyExA(
    [in]          HKEY          hKey,
    [in]          LPCSTR        lpSubKey,
    DWORD         Reserved,
    [in, optional] LPSTR        lpClass,
    [in]          DWORD         dwOptions,
    [in]          REGSAM        samDesired,
    [in, optional] const LPSECURITY_ATTRIBUTES lpSecurityAttributes,
    [out]          PHKEY         phkResult,
    [out, optional] LPDWORD      lpdwDisposition
);
```

- *hKey* (handle chiave padre): handle della chiave padre dove viene creata o aperta la nuova chiave;
- *lpSubkey* (nome della chiave): è una stringa e rappresenta la sottochiave da creare o aprire;
- *lpClass* (classe della chiave): è una stringa vuota o null;
- *dwOptions* (opzioni): opzioni aggiuntive per la creazione o apertura della chiave;
- *samDesired* (accesso desiderato): il livello di accesso desiderato della chiave;
- *lpSecurityAttributes* (attributi di sicurezza): relativi alla nuova chiave
- *phkResult* (puntatore all'handle della nuova chiave): puntatore a cui viene restituito l'handle della nuova chiave creata o aperta

Alla locazione 00401021, i parametri `lpSubKey` della funzione `RegCreateKeyExA`, vengono passati con l'istruzione 'push'. Nello specifico, la funzione modifica la chiave nella cartella "Software\\Microsoft\\Windows\\CurrentVersion", il quale contiene molte informazioni di configurazione relative alla versione attuale del sistema operativo installato.

<code>.text:00401004</code>	<code>push 0 ; lpdwDisposition</code>
<code>.text:00401006</code>	<code>lea eax, [ebp+hObject]</code>
<code>.text:00401009</code>	<code>push eax ; phkResult</code>
<code>.text:0040100A</code>	<code>push 0 ; lpSecurityAttributes</code>
<code>.text:0040100C</code>	<code>push 0F003Fh ; samDesired</code>
<code>.text:00401011</code>	<code>push 0 ; dwOptions</code>
<code>.text:00401013</code>	<code>push 0 ; lpClass</code>
<code>.text:00401015</code>	<code>push 0 ; Reserved</code>
<code>.text:00401017</code>	<code>push offset SubKey ; "SOFTWARE\\Microsoft\\Windows NT\\CurrentVe"...</code>
<code>.text:0040101C</code>	<code>push 80000002h ; hKey</code>

3. Che oggetto rappresenta il parametro alla locazione 00401017

```
.text:00401017      push    offset SubKey ; "SOFTWARE\\Microsoft\\Windows NT\\CurrentVe"...
```

Andando alla locazione 00401017, vediamo che viene passato il parametro `Subkey` che andrà a inserire nello stack, l'indirizzo di memoria della cartella "Software\\Microsoft\\Windows\\CurrentVersion". Tale indirizzo verrà utilizzato come argomento quando verrà richiamata la funzione

4. Il significato delle istruzioni comprese tra gli indirizzi 00401027 e 00401029

<code>.text:00401027</code>	<code>test eax, eax</code>
<code>.text:00401029</code>	<code>jz short loc_401032</code>

Tra gli indirizzi 00401027 e 00401029 abbiamo:

- L'istruzione `test eax, eax` esegue un'operazione di 'AND' logico tra sè stesso e il valore del registro, tuttavia non modifica il valore di `eax`, ma modifica lo Zero Flag (ZF) del registro EFLAGS.
- L'istruzione `jz short loc_401032`, effettua un salto condizionale, ciò significa che l'istruzione `jz` (jump if zero) esegue un salto alla destinazione specificata = `loc_401032` solo se lo zero flag (ZF) del registro EFLAGS = 1

```
.text:00401032 loc_401032: ; CODE XREF: sub_401000+29↑j
```


5. Con riferimento all'ultimo quesito, tradurrà il codice Assembly nel corrispondente costruito C

```
int main() {  
    int a;                //eax  
    int c;                //ecx  
    int d;                // [ebp + cbData]  
  
    if (a == 0) {  
        c = d;  
    } else {  
        a = 1;  
    }  
    return 0; }
```

6. Valutare ora la chiamata alla locazione 00401047, qual'è il valore del parametro <<ValueName>>?

Alla locazione di memoria 00401047 viene chiamata la funzione `RegSetValueExA` che permette di impostare i dati e il tipo di un valore specificato nella chiave del Registro di sistema.

```
.text:00401047      call     ds:RegSetValueExA
```

ValueName è un parametro della funzione che va a specificare il nome del valore da impostare per la chiave di registro per la funzione che andremo a creare e/o modificare. Nel nostro caso il parametro ValueName è chiamato "GinaDLL"

```
.text:0040103E      push     offset ValueName ; "GinaDLL"
```

ANALISI DINAMICA

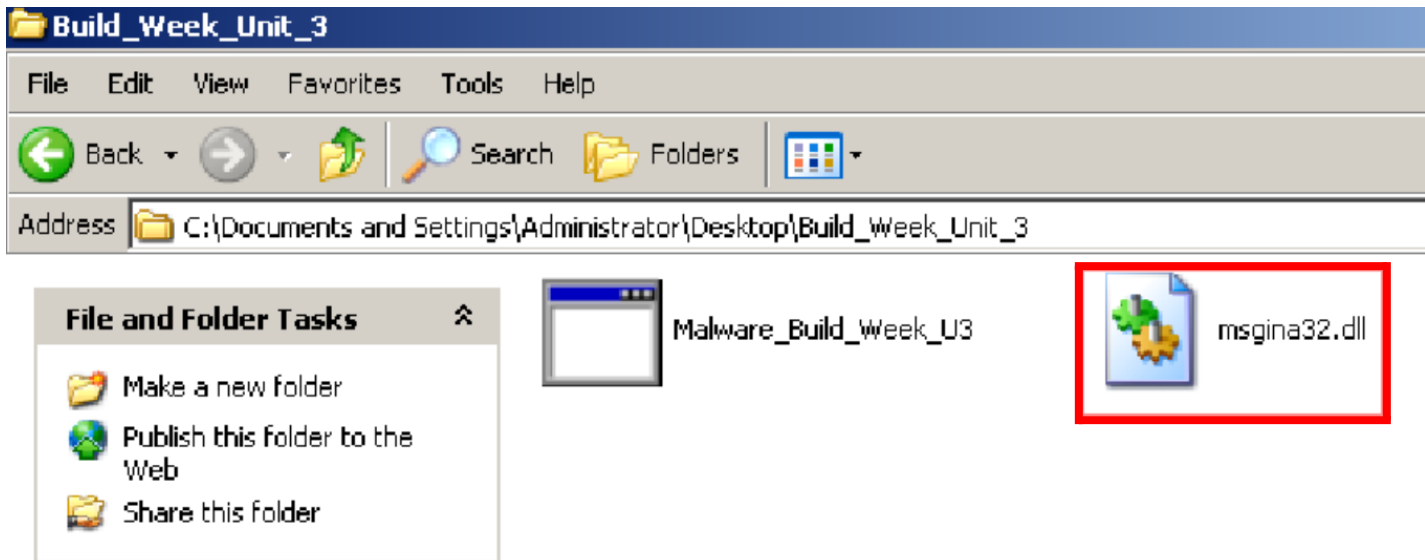
L'analisi dinamica comprende tutte quelle attività di analisi che presuppongono l'esecuzione del malware in un ambiente dedicato. L'analisi dinamica basica (in termini di processo) viene effettuata dopo l'analisi statica basica, per sopprimere i limiti dell'analisi statica ed avere una maggiore visibilità sulle attività e il comportamento del malware in esame.

Per avviare un malware basterà eseguire un doppio click sul relativo file eseguibile con formato <<.exe>>.

Per monitorare i comportamenti dei malware in esecuzione, utilizziamo un tool: Process Monitor/procmon, un tool avanzato per Windows che permette di monitorare i processi e i thread attivi, l'accesso ai file e le chiamate di sistema effettuare su un sistema operativo.

Per l'analisi dinamica andremo a rimuovere tutti i filtri preimpostati, per far sì che Process Monitor venga avviato senza nessun filtro.

Nella cartella dove è contenuto il malware è stato creato il file “msgina32.dll”



Avviando la scansione avremo come risultato una serie di operazioni, mirate a creare una mappatura del sistema e dele directory.

Tra queste possiamo notare CreateFile, CloseFile...

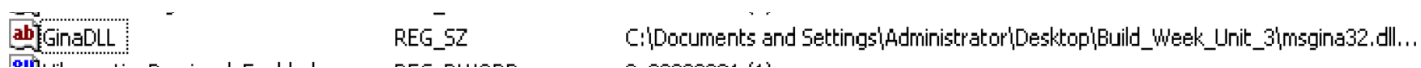
1592	CreateFile	C:\WINDOWS\Prefetch\MALWARE_BUILD_WEEK_U3.EXE-0E171D0F.pf	SUCCESS
1592	QueryStandardInformationFile	C:\WINDOWS\Prefetch\MALWARE_BUILD_WEEK_U3.EXE-0E171D0F.pf	SUCCESS
1592	ReadFile	C:\WINDOWS\Prefetch\MALWARE_BUILD_WEEK_U3.EXE-0E171D0F.pf	SUCCESS
1592	CloseFile	C:\WINDOWS\Prefetch\MALWARE_BUILD_WEEK_U3.EXE-0E171D0F.pf	SUCCESS
1592	CreateFile	C:	SUCCESS
1592	QueryInformationVolume	C:	SUCCESS

Andando nello specifico, andiamo a filtrare sul registro di sistema attivo e possiamo osservare le operazioni RegCreateKey e RegSetValue

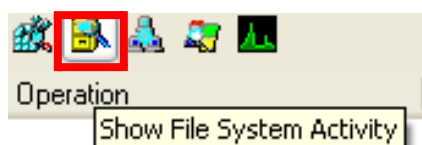


1592	RegCreateKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
1592	RegSetValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\GinaDLL

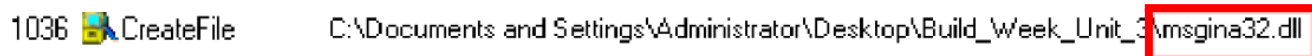
- RegCreateKey è stata utilizzata per modificare la chiave di registro che conteneva le informazioni necessarie per effettuare l'accesso e l'autenticazione degli utenti.
- RegSetValue vediamo che il malware ha modificato la chiave inserendo un nuovo valore, che corrisponde al percorso del file **“msgina32.dll”**: infatti aprendo il registro di sistema con il comando regedit, troviamo il nuovo valore inserito:



Adesso andiamo a filtrare la nostra ricerca per 'File system'



Così facendo vedremo le attività di mappatura viste anche in precedenza, notiamo però un'attività di 'CreateFile' nel percorso del malware (msgina32) che prima dell'avvio del malware non esisteva



MALWARE ANALYSIS

Cosa notate all'interno della cartella dove è situata l'eseguibile del Malware? Spiegate cosa è avvenuto, unendo le evidenze che avete raccolto finora per rispondere alla domanda.

Analizzate ora i risultati di Processo Monitor. Fate click su 'ADD' poi su 'Apply'

Il malware va a creare un file chiamato **“msgina32.dll”** sfruttando la chiave di autenticazione Windows chiamata GinaDLL, andando a impostare il suo percorso di creazione e sovrascrivendolo, facendo sì che il malware ottenga l'accesso automatico della macchina e/o permettendo azioni non autorizzate.

Le informazioni raccolte infatti, ci inducono a presumere che il sistema sia stato compromesso da un malware di tipo 'dropper'. Lo vediamo in particolare con le modifiche al Registro di sistema, dove notiamo che il malware stia cercando di stabilire una presenza assidua sulla macchina, garantendo così l'esecuzione automatica all'avvio del sistema operativo.