

# Project Overview

**Buonanno Manuel**  
**29/03/2024**



## Traccia

Con riferimento al file Malware\_U3\_W2\_L5 presente all'interno della cartella «Esercizio\_Pratico\_U3\_W2\_L5 » sul desktop della macchina virtuale dedicata per l'analisi dei malware, rispondere ai seguenti quesiti:

1. Quali librerie vengono importate dal file eseguibile?
2. Quali sono le sezioni di cui si compone il file eseguibile del malware?

Con riferimento alla figura, risponde ai seguenti quesiti:

3. Identificare i costrutti noti (creazione dello stack, eventuali cicli, altri costrutti).
4. Ipotizzare il comportamento della funzionalità implementata 5. BONUS fare tabella con significato delle singole righe di codice assembly.

```
push    ebp
mov     ebp, esp
push    ecx
push    0          ; dwReserved
push    0          ; lpdwFlags
call    ds:InternetGetConnectedState
mov     [ebp+var_4], eax
cmp     [ebp+var_4], 0
jz     short loc_40102B
```

```
[NUL]
push    offset aSuccessInterne ; "Success: Internet Connection\n"
call    sub_40117F
add     esp, 4
mov     eax, 1
jmp     short loc_40103A
```

```
[NUL]
loc_40102B:           ; "Error 1.1: No Internet\n"
push    offset aError1_1NoInte
call    sub_40117F
add     esp, 4
xor     eax, eax
```

```
[NUL]
loc_40103A:
mov     esp, ebp
pop    ebp
retn
sub_401000 endp
```

## Analisi statica basica

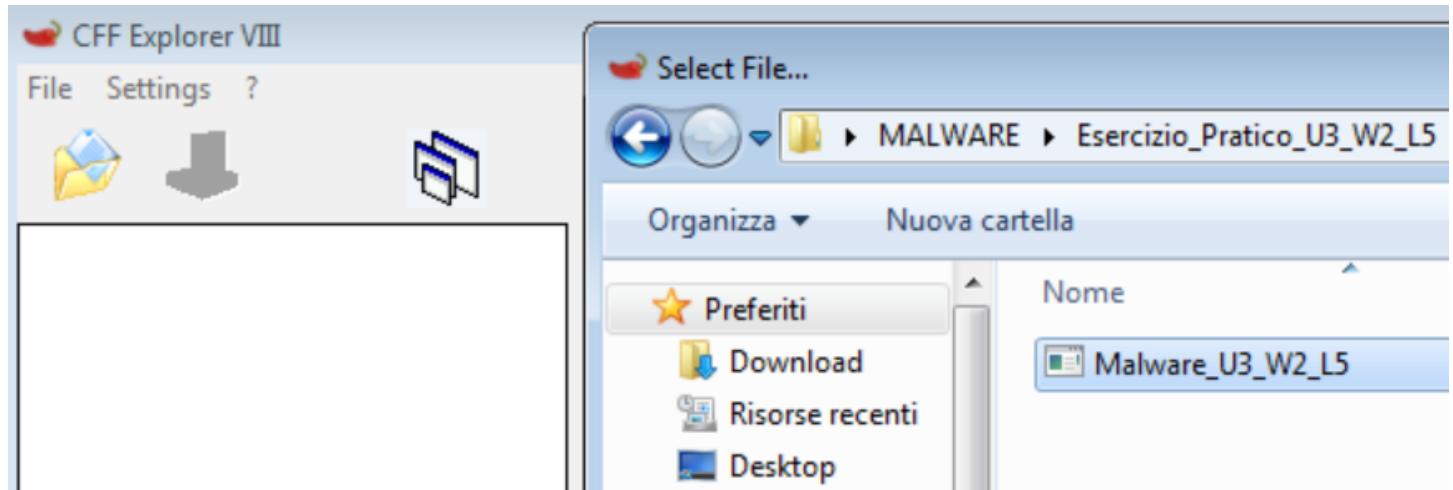
L'analisi statica basica consiste nell'esaminare un eseguibile senza vedere le istruzioni che lo compongono. Lo scopo è quello di confermare se un dato file è malevolo e fornire informazioni generiche circa le sue funzionalità. Questa metodologia è sicuramente la più intuitiva e semplice da mettere in pratica, ma risulta anche essere la più inefficiente soprattutto contro malware sofisticati.

Windows utilizza per la maggior parte dei file eseguibili il formato PE, PortableExecutable. Il formato PE al suo interno contiene delle informazioni necessarie al sistema operativo per capire come gestire il codice del file, come ad esempio le librerie e funzioni. Quando un programma ha bisogno di una funzione “chiama” una libreria al cui interno è definita la funzione necessaria

Oltre alle funzioni importate, un file eseguibile può esportare funzioni. Ovvero, può mettere a disposizione di altri programmi o dell'utente delle funzioni da “chiamare”. L'header del formato PE contiene anche un elenco delle funzioni esportate da un eseguibile. Per controllare le funzioni importate ed esportate da un malware, possiamo utilizzare il tool CFF Explorer.

## CFF Explorer

Sul desktop della nostra macchina virtuale windows 7 troveremo due cartelle: una con diversi gli eseguibili di diversi malware e una con i software per analizzarli.

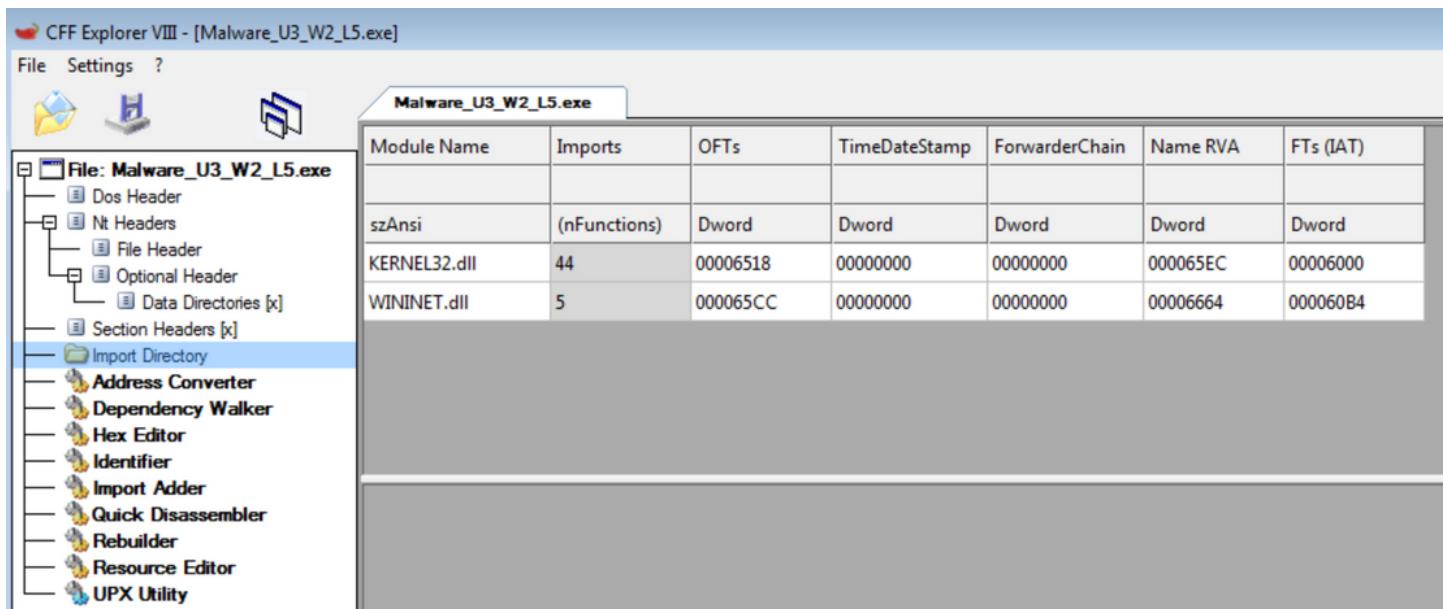


Avviamo quindi CFF Explorer e ci ritroveremo la seguente interfaccia. Cliccando sulla cartella gialla nell'angolo andremo a caricare il file «Esercizio\_Pratico\_U3\_W2\_L5 »

### CFF Explorer: Import Directory

Possiamo notare come CFF Explorer analizza tutto il file che abbiamo caricato restituendoci molte informazioni su esso.

Per controllare le librerie e le funzioni importate, ci spostiamo su «import directory» nel menù a sinistra. Il pannello darà informazioni sulle librerie importate dall'eseguibile.



In questo caso il malware che abbiamo analizzato importa 2 librerie:

- **Kernel32.dll**, contiene le funzioni principali per interagire con il sistema operativo, ad esempio: manipolazione dei file, la gestione della memoria;
- **Wininet.dll**, contiene le funzioni per l'implementazione di alcuni protocolli di rete come HTTP, FTP, NTP.

Il pannello inferiore mostrerà una lista delle funzioni richieste all'interno della libreria selezionate, quindi cliccare su KERNEL32.dll e WININET.dll.

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
000065E4	000065E4	0296	Sleep
00006940	00006940	027C	SetStdHandle
0000692E	0000692E	0156	GetStringTypeW
0000691C	0000691C	0153	GetStringTypeA
0000690C	0000690C	01C0	LCMapStringW
000068FC	000068FC	01BF	LCMapStringA

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	<a href="#">szAnsi</a>
00006640	00006640	0071	<a href="#">InternetOpenUrlA</a>
0000662A	0000662A	0056	<a href="#">InternetCloseHandle</a>
00006616	00006616	0077	<a href="#">InternetReadFile</a>
000065FA	000065FA	0066	<a href="#">InternetGetConnectedState</a>
00006654	00006654	006F	<a href="#">InternetOpenA</a>

Le funzioni di un malware possono fornire indicazioni significative sulle sue capacità e sui suoi obiettivi. Ecco cosa si potrebbe dedurre dalle funzioni menzionate:

- **szAnsi, GetStringTypeW, GetStringTypeA, LCMapStringW, LCMapStringA:** queste funzioni coinvolgono la gestione e la manipolazione delle stringhe di testo. La presenza di tali funzioni potrebbe indicare che il malware sta cercando di ottenere o manipolare informazioni testuali presenti nel sistema infetto.
- **Sleep:** questa funzione è comunemente utilizzata per introdurre ritardi nel codice del malware, spesso utilizzati per eludere la rilevazione e per sincronizzare azioni specifiche. Il suo utilizzo potrebbe indicare una strategia per mascherare l'attività del malware e per evitare l'individuazione da parte dei sistemi di sicurezza.
- **SetStdHandle:** questa funzione può essere utilizzata per modificare gli handle di file standard del sistema operativo. Potrebbe essere utilizzato per modificare il comportamento dell'I/O del malware, indirizzando l'output a risorse specifiche o modificando la gestione dei file standard del sistema.
- **InternetOpenUrlA, InternetCloseHandle, InternetReadFile, InternetGetConnectedState, InternetOpenA:** queste funzioni coinvolgono l'interazione con Internet. Il loro utilizzo potrebbe indicare che il malware tenta di stabilire una connessione a un server remoto per scaricare ulteriori componenti, inviare dati rubati o ricevere comandi da un server.

Complessivamente, la presenza di queste funzioni può suggerire che il malware è progettato per svolgere attività dannose come la raccolta di

informazioni, la comunicazione con server remoti o la manipolazione dei file di sistema.

## CFF Explorer: Section Headers

Per controllare le sezioni di un file eseguibile spostiamoci nel pannello a sinistra nella sezione «Section Headers». Il pannello a destra mostrerà le informazioni circa le sezioni di cui si compone l'eseguibile.

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N...	Linenumbers ...	Characteristics
Byte[8]	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword	60000020
.text	00004A78	00001000	00005000	00001000	00000000	00000000	0000	0000	40000040
.rdata	0000095E	00006000	00001000	00006000	00000000	00000000	0000	0000	C0000040
.data	00003F08	00007000	00003000	00007000	00000000	00000000	0000	0000	

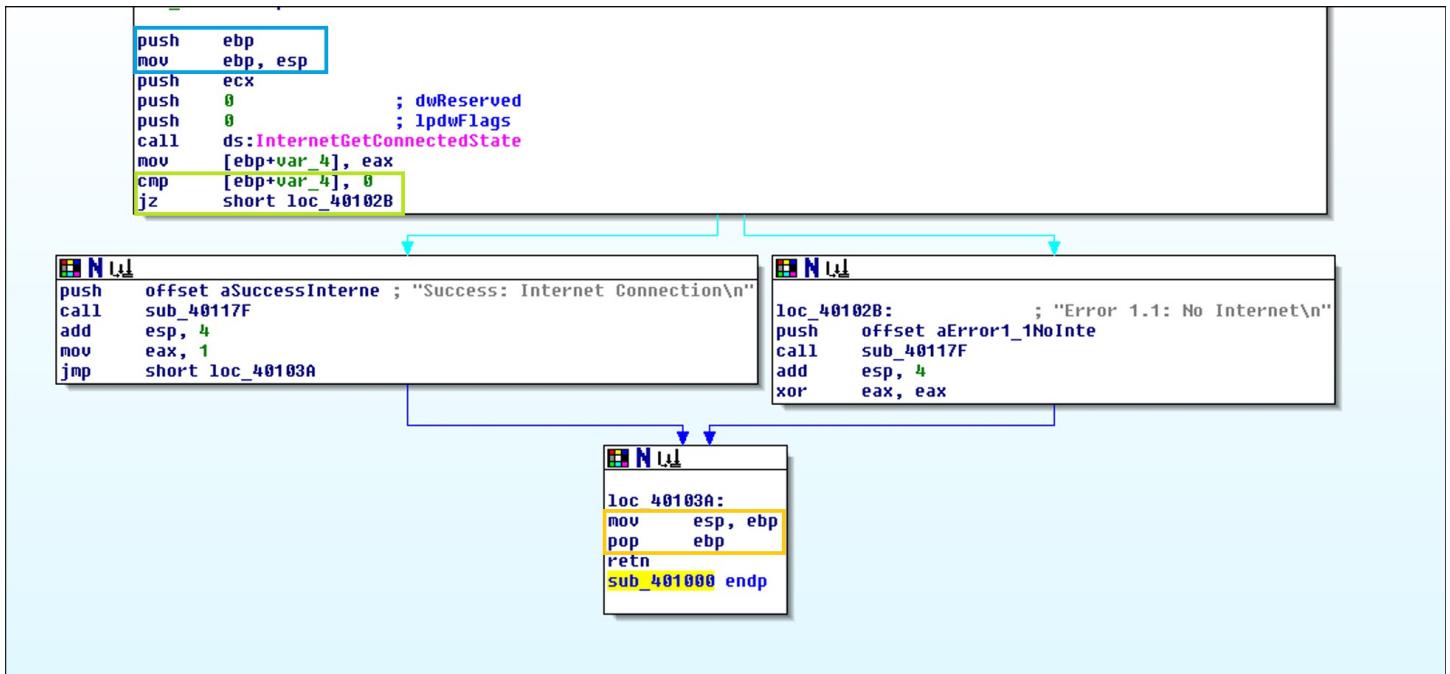
La figura riporta non solo il nome delle sezioni ma anche altre importanti informazioni, come ad esempio:

- **Virtual size:** indica lo spazio allocato per la sezione durante il processo di caricamento dell'eseguibile in memoria.
- **Rawsize:** indica lo spazio occupato dalla sezione quando è sul disco.

L'header del formato PE fornisce molte altre informazioni importanti oltre alle funzioni/librerie importate ed esportate, come ad esempio le sezioni di cui si compone il software. In questo caso possiamo notare:

- **.text:** contiene le istruzioni 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.
- **.data:** contiene tipicamente i dati / le variabili globali del programma eseguibile, che devono essere disponibili da qualsiasi parte del programma.

## Assembly: Costrutti



### Riquadro blu: creazione stack

Queste istruzioni vengono utilizzate per creare un nuovo frame di stack. <<push ebp>> salva il valore corrente del registro base dello stack (ebp) nello stack, e <<mov ebp, esp>> imposta il registro base dello stack al valore corrente del puntatore allo stack (esp).

### Riquadro verde: condizione if

Esegue una verifica sul valore restituito dalla funzione <<InternetGetConnectedState>>, che presumibilmente indica lo stato della connessione Internet.

### Riquadro giallo: elimino stack

Queste istruzioni vengono utilizzate per eliminare il frame di stack corrente. <<mov esp, ebp>> imposta il puntatore allo stack al valore corrente del registro base dello stack (ebp), e <<pop ebp>> ripristina il valore precedente del registro base dello stack rimuovendolo dallo stack.

## Assembly: Comportamento

Il codice sembra essere una routine che controlla lo stato della connessione Internet e stampa un messaggio appropriato a seconda dello stato della connessione. Utilizza la funzione <<InternetGetConnectedState>> per verificare la connessione e quindi stampa un messaggio di successo

o errore.

## Assembly: Analisi codice

- push ebp: Salva il valore corrente del puntatore alla base dello stack.
- mov ebp, esp: Imposta il puntatore alla base dello stack al valore corrente del puntatore allo stack.
- push ecx: Salva il valore corrente del registro ECX nello stack.
- push 0: Pone 0 sullo stack, probabilmente come parametro per una funzione.
- push 0: Pone un altro 0 sullo stack, probabilmente un altro parametro per la funzione.
- call ds:InternetGetConnectedState: Chiama la funzione InternetGetConnectedState per controllare lo stato della connessione Internet.
- mov [ebp+var\_4], eax: Salva il valore restituito dalla funzione InternetGetConnectedState nello spazio di memoria locale var\_4.
- cmp [ebp+var\_4], 0: Compara il valore restituito con 0 per vedere se c'è una connessione Internet.
- jz short loc\_40102B: Salta a loc\_40102B se non c'è connessione Internet.
- push offset ASuccessInterne: Pone l'offset di una stringa "Success: Interne Connection\n" nello stack come parametro per la funzione di stampa.
- call sub\_40117F: Chiama una subroutine che stampa il messaggio di successo.
- add esp, 4: Ripristina lo stack dopo aver rimosso il parametro.
- mov eax, 1: Imposta il registro eax a 1, presumibilmente per indicare il successo.
- jmp loc\_40103A: Salta all'indirizzo loc\_40103A.
- loc\_40102B:: Etichetta per la gestione dell'errore quando non c'è connessione Internet.
- push offset aError1\_1Nolnte: Pone l'offset di una stringa "Error 1.1: No internet\n" nello stack come parametro per la funzione di stampa.
- call sub\_40117F: Chiama una subroutine che stampa il messaggio di errore.
- add esp, 4: Ripristina lo stack dopo aver rimosso il parametro.
- xor eax, eax: Imposta eax a 0, presumibilmente per indicare un errore.
- loc\_40103A:: Etichetta per il completamento della routine.
- mov esp, ebp: Ripristina il puntatore allo stack.
- pop ebp: Ripristina il valore del puntatore alla base dello stack.

- retn: Restituisce il controllo al chiamante.
- sub\_401000 endp : fine blocco subroutine.

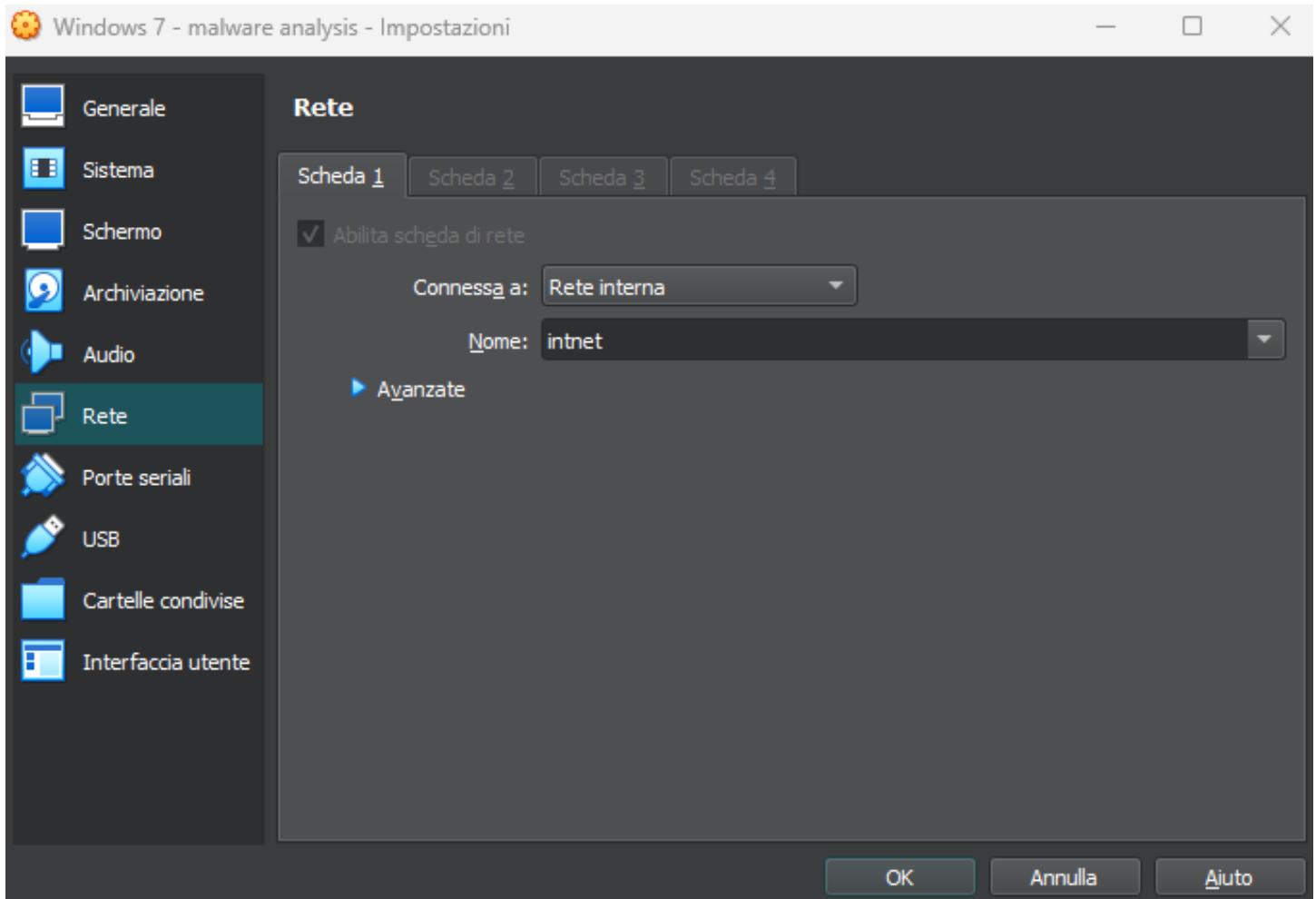
## Extra: Analisi dinamica

Per un'analisi efficace i risultati delle analisi statiche devono essere poi confermate dai risultati delle analisi dinamiche. L'analisi dinamica basica presuppone l'esecuzione del malware in modo tale da osservare il suo comportamento sul sistema infetto al fine di rimuovere l'infezione. I malware devono essere eseguiti in ambiente sicuro e controllato in modo tale da eliminare ogni rischio di arrecare danno a sistemi o all'intera rete.

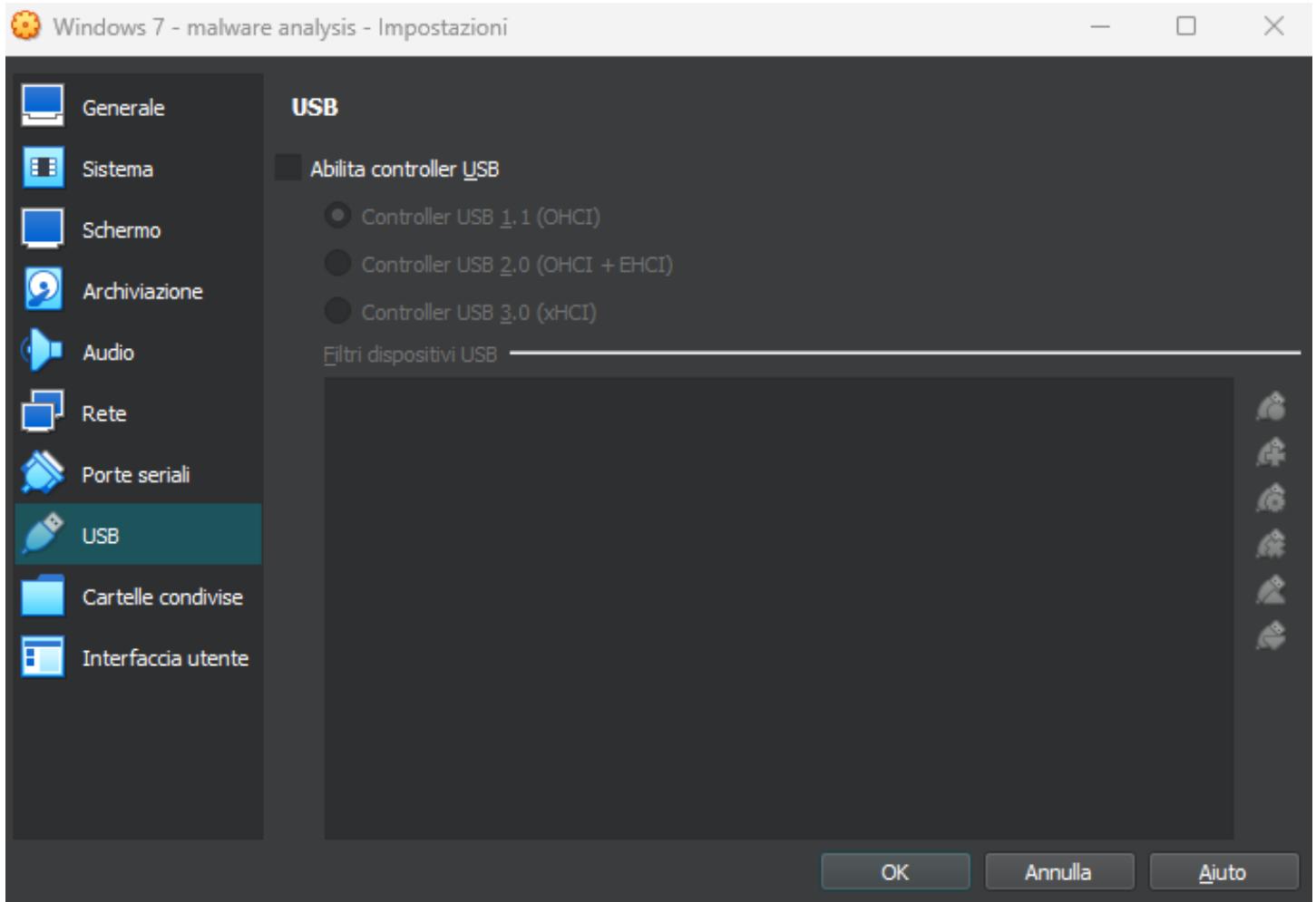
## Configurazione macchina virtuale

Prima di eseguire un ananlisi dinamica dobbiamo adottare delle pratiche per rendere sicuro il nostro ambiente.

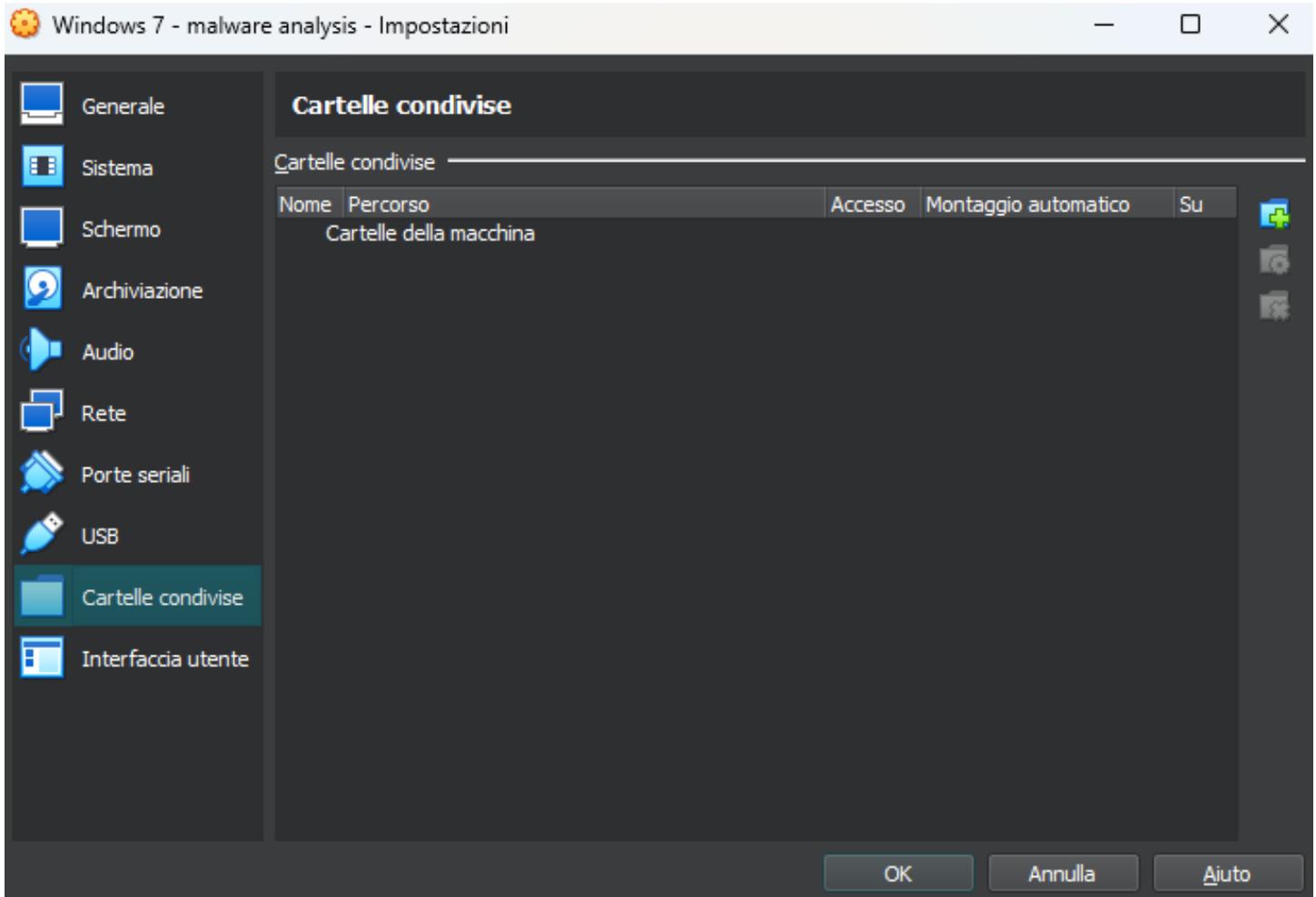
- Configurazione schede di rete: l'ambiente di test non deve avere accesso diretto ad Internet e preferibilmente nemmeno accesso ad altre macchine sulla rete. La configurazione ideale è eliminare le interfacce di rete durante l'analisi statica;
- Abilitare un'interfaccia di rete interna (su VirtualBox viene chiamata «rete interna») per l'analisi dinamica. Questa impostazione è necessaria per monitorare il traffico che genera potenzialmente il malware.



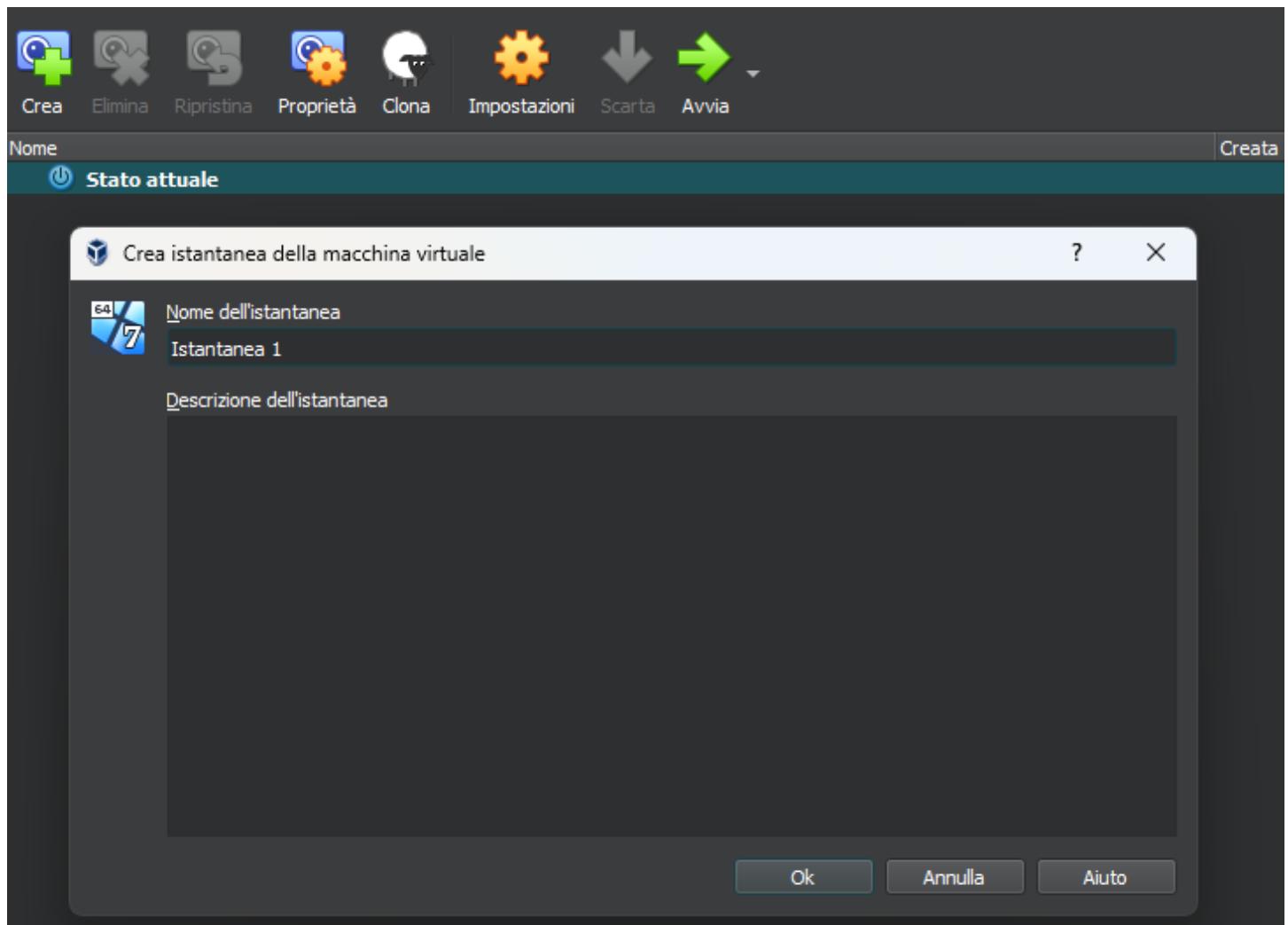
- Dispositivi USB: quando un dispositivo USB viene collegato alla macchina fisica, esso può essere riconosciuto anche dall'ambiente di test. Al fine di evitare questo comportamento, è buona pratica non abilitare o disabilitare il controller USB. Infatti, il malware potrebbe utilizzare il dispositivo USB per propagarsi poi sulla vostra macchina fisica. La figura di fianco mostra l'impostazione in VirtualBox, «abilita controller USB» NON deve essere abilitato.



- Cartelle condivise: stesso discorso può essere per le cartelle condivise tra la vostra macchina reale ed il laboratorio virtuale. Potrebbero essere utilizzate dal malware per propagarsi al di fuori del laboratorio causando danni alla vostra macchina e alle macchine sulla vostra rete domestica. Di conseguenza, è consigliato non condividere cartelle tra host e guest.



- Creare delle istantanee: Una buona pratica è creare delle istantanee della macchina virtuale nel suo stato iniziale, prima di iniziare tutte le analisi, in modo tale da ripristinarlo qualora ce ne fosse bisogno. Per creare un'istantanea, cliccate su «crea» (1), poi su OK (2) dopo aver inserito un nome ed una descrizione facoltativa.  
Se l'ambiente virtuale dovesse risultare compromesso, potete ripristinare l'istantanea cliccando sull'icona «ripristina» dopo averla selezionata dalla lista. Assicuratevi quindi di avviare la macchina avendo cura di selezionare «stato attuale» dalla lista.

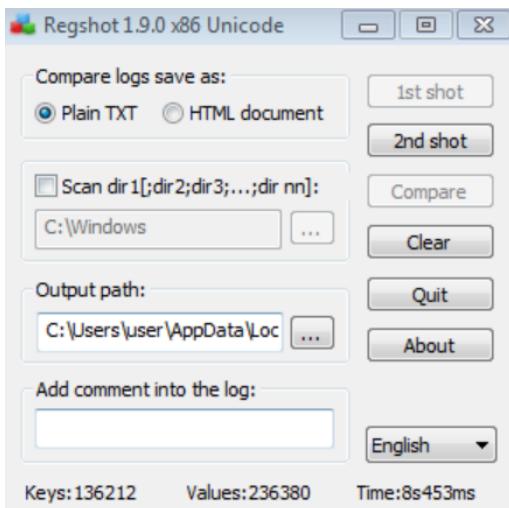


## Programmi utilizzati

- RegShot: permette di visualizzare possibili modifiche prima e dopo aver lanciato un malware.
- ApateDNS: utilizzato per simulare un server DNS e se propriamente configurato, può intercettare tutte le richieste effettuate dai malware verso i domini Internet.
- Procmon: è un tool vanzato 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.

## Regshot 1

Prima dell'esecuzione del malware eseguiamo uno "shot" in modo da controllare le differenze prima e dopo.



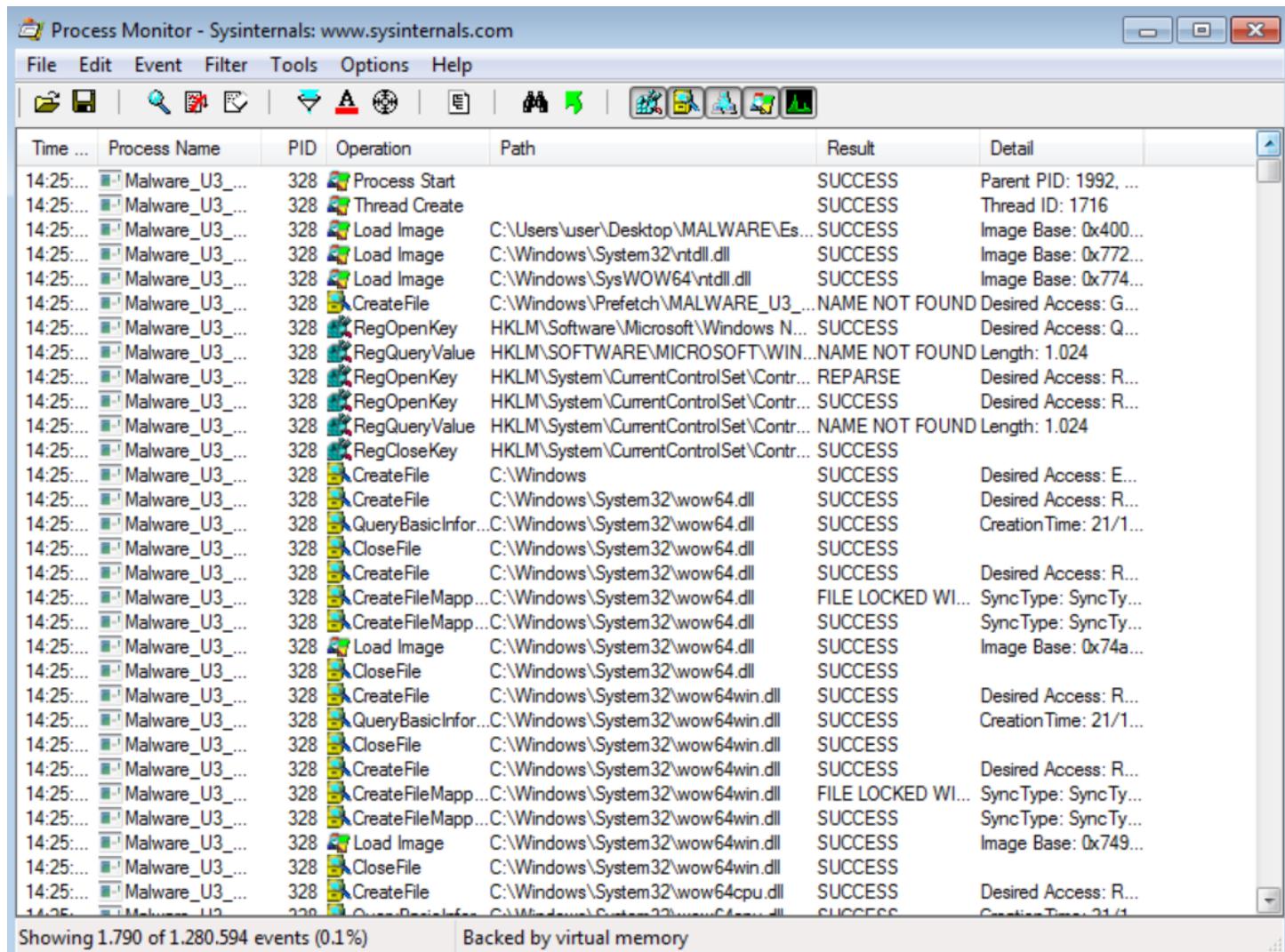
## Apate DNS

Inserito l'ip della macchina in uso clicchiamo su Start server. Avviamo il malware e notiamo che non otteniamo nessuna risposta, il che significa che questo malware non usa in alcun modo la rete.

```
[+] Using 192.168.1.50 as return DNS IP!
[+] DNS set to 127.0.0.1 on Scheda desktop Intel(R) PRO/1000 MT.
[+] Sending valid DNS response of first request.
[+] Server started at 14:24:02 successfully.
[-] Already initiated...
```

## Procmon

Filtriamo i risultati inserendo inserendo il *process name* del malware mentre attiviamo tutte le tipologie di catture.



The screenshot shows the Process Monitor interface with the title bar "Process Monitor - Sysinternals: www.sysinternals.com". The menu bar includes File, Edit, Event, Filter, Tools, Options, and Help. Below the menu is a toolbar with various icons. The main window displays a table of events. The columns are: Time ..., Process Name, PID, Operation, Path, Result, and Detail. The table lists numerous events for process "Malware\_U3\_...".

Time ...	Process Name	PID	Operation	Path	Result	Detail
14:25:	Malware_U3_...	328	Process Start		SUCCESS	Parent PID: 1992, ...
14:25:	Malware_U3_...	328	Thread Create		SUCCESS	Thread ID: 1716
14:25:	Malware_U3_...	328	Load Image	C:\Users\user\Desktop\MALWARE\Es...	SUCCESS	Image Base: 0x400...
14:25:	Malware_U3_...	328	Load Image	C:\Windows\System32\ntdll.dll	SUCCESS	Image Base: 0x772...
14:25:	Malware_U3_...	328	Load Image	C:\Windows\SysWOW64\ntdll.dll	SUCCESS	Image Base: 0x774...
14:25:	Malware_U3_...	328	CreateFile	C:\Windows\Prefetch\MALWARE_U3_... NAME NOT FOUND	Desired Access: G...	
14:25:	Malware_U3_...	328	RegOpenKey	HKEY_LOCAL_MACHINE\Software\Microsoft\Windows N...	SUCCESS	Desired Access: Q...
14:25:	Malware_U3_...	328	RegQueryValue	HKEY_LOCAL_MACHINE\Software\Microsoft\Windows N...	NAME NOT FOUND	Length: 1.024
14:25:	Malware_U3_...	328	RegOpenKey	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Contr...	REPARSE	Desired Access: R...
14:25:	Malware_U3_...	328	RegOpenKey	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Contr...	SUCCESS	Desired Access: R...
14:25:	Malware_U3_...	328	RegQueryValue	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Contr...	NAME NOT FOUND	Length: 1.024
14:25:	Malware_U3_...	328	RegCloseKey	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Contr...	SUCCESS	
14:25:	Malware_U3_...	328	CreateFile	C:\Windows	SUCCESS	Desired Access: E...
14:25:	Malware_U3_...	328	CreateFile	C:\Windows\System32\wow64.dll	SUCCESS	Desired Access: R...
14:25:	Malware_U3_...	328	QueryBasicInfor...	C:\Windows\System32\wow64.dll	SUCCESS	CreationTime: 21/1...
14:25:	Malware_U3_...	328	CloseFile	C:\Windows\System32\wow64.dll	SUCCESS	
14:25:	Malware_U3_...	328	CreateFile	C:\Windows\System32\wow64.dll	SUCCESS	Desired Access: R...
14:25:	Malware_U3_...	328	CreateFileMapp...	C:\Windows\System32\wow64.dll	FILE LOCKED WI...	SyncType: SyncTy...
14:25:	Malware_U3_...	328	CreateFileMapp...	C:\Windows\System32\wow64.dll	SUCCESS	SyncType: SyncTy...
14:25:	Malware_U3_...	328	Load Image	C:\Windows\System32\wow64.dll	SUCCESS	Image Base: 0x74a...
14:25:	Malware_U3_...	328	CloseFile	C:\Windows\System32\wow64.dll	SUCCESS	
14:25:	Malware_U3_...	328	CreateFile	C:\Windows\System32\wow64win.dll	SUCCESS	Desired Access: R...
14:25:	Malware_U3_...	328	QueryBasicInfor...	C:\Windows\System32\wow64win.dll	SUCCESS	CreationTime: 21/1...
14:25:	Malware_U3_...	328	CloseFile	C:\Windows\System32\wow64win.dll	SUCCESS	
14:25:	Malware_U3_...	328	CreateFile	C:\Windows\System32\wow64win.dll	SUCCESS	Desired Access: R...
14:25:	Malware_U3_...	328	CreateFileMapp...	C:\Windows\System32\wow64win.dll	FILE LOCKED WI...	SyncType: SyncTy...
14:25:	Malware_U3_...	328	CreateFileMapp...	C:\Windows\System32\wow64win.dll	SUCCESS	SyncType: SyncTy...
14:25:	Malware_U3_...	328	Load Image	C:\Windows\System32\wow64win.dll	SUCCESS	Image Base: 0x749...
14:25:	Malware_U3_...	328	CloseFile	C:\Windows\System32\wow64win.dll	SUCCESS	
14:25:	Malware_U3_...	328	CreateFile	C:\Windows\System32\wow64cpu.dll	SUCCESS	Desired Access: R...
14:25:	Malware_U3_...	328	QueryBasicInfor...	C:\Windows\System32\CPU.dll	SUCCESS	CreationTime: 21/1...

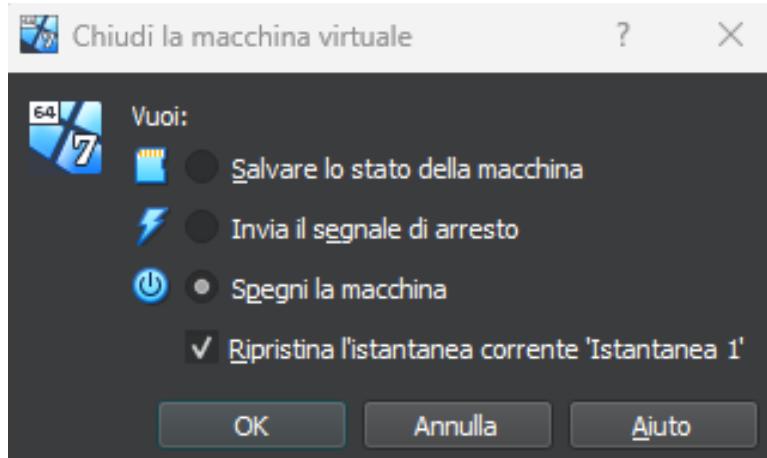
Notiamo che sono stati modificati dei file all'interno del file system e delle chiavi di registro.

## Regshot 2

Eseguiamo adesso un secondo “shot” e andiamo a comparare i risultati. Come notiamo ci sono state delle modifiche tra le quali chiavi aggiunte, modificate ed eliminate.

# Istantanea

Chiudiamo infine la macchina ripristinando l'istantanea creata in precedenza.





EPICODE