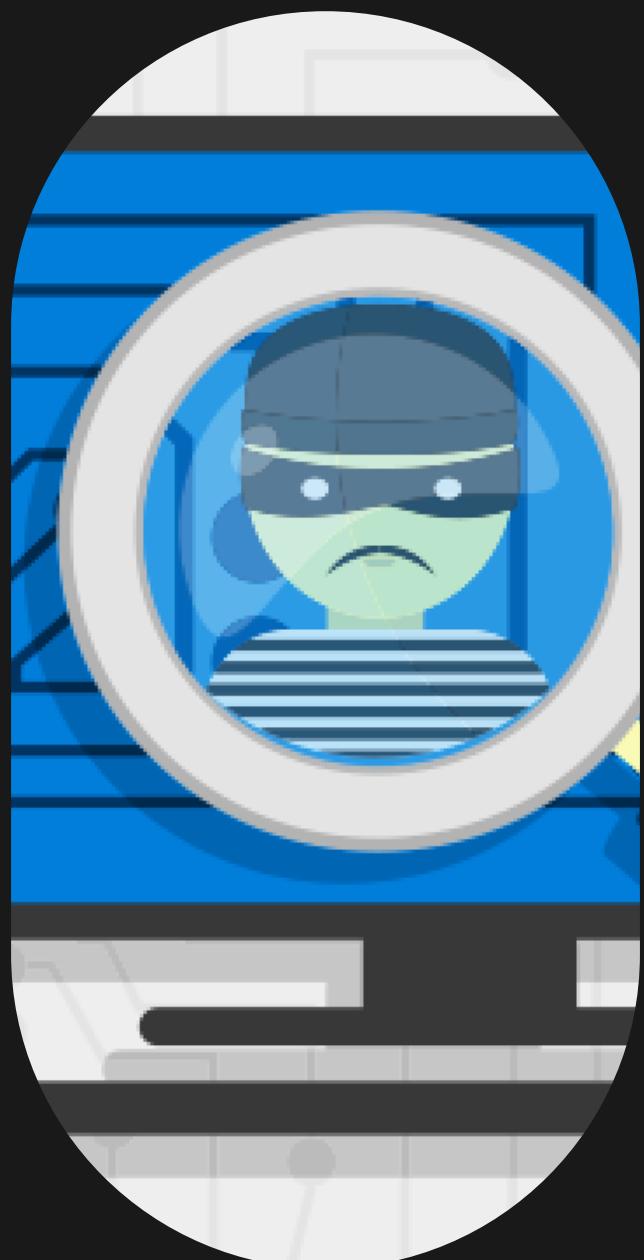


Malware Analysis e Reverse engineering

Prepared by

Alessandro Bossi, CS 02-23

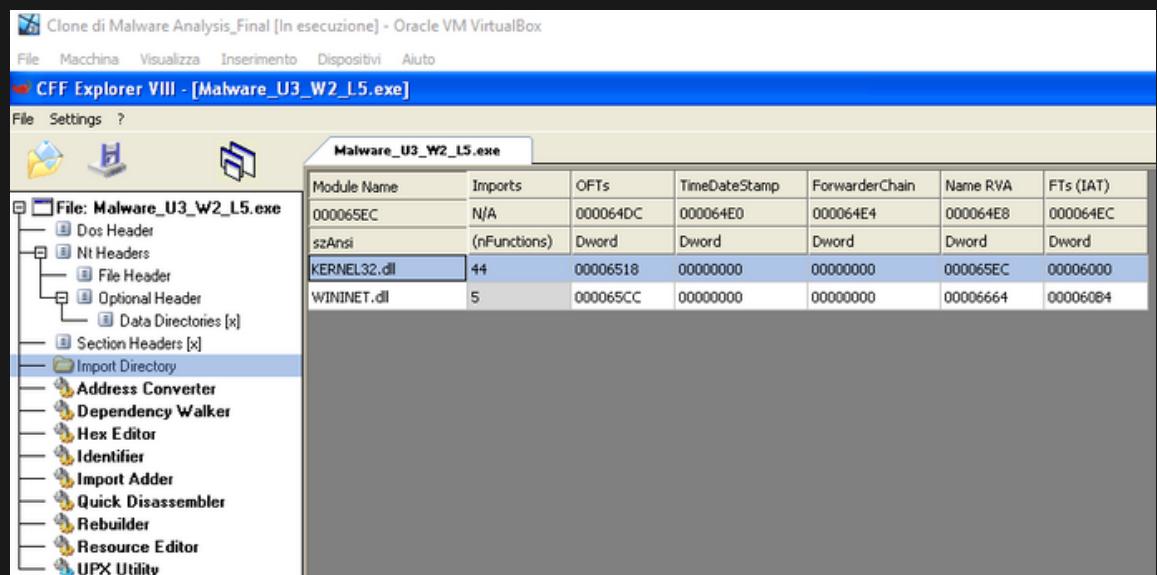


Content Highlights

- Analisi statica base
- assembly
- bonus: analisi dinamica e statica base



la prima parte dell'esercizio consiste nell' analisi del file "Malware_U3_W2_L5" presente all'interno della macchina virtuale Windows XP dedicata all'analisi malware. Procedo quindi con l'avvio dell'app **CFF Explorer** ed inizio un'analisi statica base per recuperare le **librerie** che sono state importate dal file e le **sezioni** di cui è composto.



Parto dall' analisi delle librerie importate e mi sposto nella sezione **Import Directory**, dove vedo che ci sono due librerie: **Kernel32.dll** e **Wininet.dll**. La prima è una libreria a collegamento dinamico (DLL) che fa parte del sistema operativo Windows. Contiene funzioni e risorse essenziali utilizzate da vari programmi e applicazioni eseguite su Windows.

La seconda è una libreria a collegamento dinamico (DLL) utilizzata dal sistema operativo Windows per fornire funzionalità di rete e accesso a Internet alle applicazioni, fornisce una serie di funzioni che consentono alle applicazioni di comunicare con i server su Internet, inviare richieste HTTP, scaricare file, gestire cookie, eseguire l'autenticazione e altro ancora.

come visto dallo screenshot precedente ogni libreria importa delle funzioni, procedo quindi alla descrizione di alcune delle più importanti per ogni libreria.

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
000068E6	000068E6	01E4	MultiByteToWideChar
00006670	00006670	00CA	GetCommandLineA
00006682	00006682	0174	GetVersion
00006690	00006690	007D	ExitProcess
0000669E	0000669E	029E	TerminateProcess
000066B2	000066B2	00F7	GetCurrentProcess
000066C6	000066C6	02AD	UnhandledExceptionFilter
000066E2	000066E2	0124	GetModuleFileNameA
000066F8	000066F8	00B2	FreeEnvironmentStringsA
00006712	00006712	00B3	FreeEnvironmentStringsW

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
00006810	00006810	02BF	VirtualFree
0000681E	0000681E	019F	HeapFree
0000682A	0000682A	022F	RtlUnwind
00006836	00006836	02DF	WriteFile
00006842	00006842	0199	HeapAlloc
0000684E	0000684E	00BF	GetCPInfo
0000685A	0000685A	00B9	GetACP
00006864	00006864	0131	GetOEMCP
00006870	00006870	02BB	VirtualAlloc
00006880	00006880	01A2	HeapReAlloc
0000688E	0000688E	013E	GetProcAddress
000068A0	000068A0	01C2	LoadLibraryA
000068B0	000068B0	011A	GetLastError
000068C0	000068C0	00AA	FlushFileBuffers
000068D4	000068D4	026A	SetFilePointer
00006950	00006950	001B	CloseHandle

Kernel32.dll importa 44 funzioni tra queste possiamo trovare:

- 1)Funzioni di gestione della memoria:
VirtualAlloc, **VirtualFree**, ecc., utilizzate per allocare, deallocare e gestire la memoria.
- 2)Funzioni di processo e thread: **CreateProcess**, **TerminateProcess**, utilizzate per creare e gestire processi.
- 3)Funzioni di operazioni sui file: **ReadFile**, **CloseHandle**, , utilizzate per eseguire operazioni relative ai file, come aprire, leggere, scrivere e chiudere file.

Wininet.dll invece importa 5 funzioni:

- 1) **InternetOpen**: Apre una connessione Internet per un'applicazione e restituisce un handle che rappresenta la connessione.
- 2) **InternetOpenUrl**: Apre una connessione a un URL specificato e restituisce un handle per il file o la risorsa richiesta.
- 3) **InternetReadFile**: Legge i dati dal file o dalla risorsa scaricata da Internet.
- 4) **InternetGetConnectedState**: consente di verificare lo stato della connessione Internet in un determinato momento.
- 5) **InternetCloseHandle**: Chiude una connessione Internet o un handle aperto precedentemente.

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
00006640	00006640	0071	InternetOpenUrlA
0000662A	0000662A	0056	InternetCloseHandle
00006616	00006616	0077	InternetReadFile
000065FA	000065FA	0066	InternetGetConnectedState
00006654	00006654	006F	InternetOpenA

in conclusione di questa prima parte possiamo ipotizzare che il malware analizzato sia in grado di effettuare operazioni direttamente all'interno del pc infetto, cosa che potrebbe portare ad una privilege escalation oltre che ad altri danni. Inoltre è in grado di effettuare e gestire connessioni ad internet, rendendolo molto pericoloso da un punto di vista di livello di diffusione e fuga di dati verso l'esterno.

la seconda parte dell'esercizio invece consiste nell'analizzare un codice assembly, individuare i costrutti noti ed ipotizzare il comportamento della funzionalità implementata.



```
push    ebp  
mov    ebp, esp
```

creazione dello stack



```
push    ecx  
push    0          ; dwReserved  
push    0          ; lpdwFlags  
call    ds:InternetGetConnectedState
```

chiamata alla funzione `InternetGetConnectedState`, i parametri vengono passati sullo stack tramite le istruzioni push



mov [ebp+var_4], eax

Copia il risultato della chiamata alla funzione nella variabile locale [ebp+var_4]



```
cmp    [ebp+var_4], 0  
jz     short loc 40102B
```

ciclo IF



```
push    offset aSuccessInternet ; "Success: Internet Connection\n"
call    sub_40117F
add    esp, 4
mov    eax, 1
jmp    short loc_401030
```

se la condizione precedente risulta essere vera, chiama la subroutine `sub_40117F` ed esegue l'output della stringa che è stata pushata "Success: Internet Connection\n", pulisce lo stack, imposta eax ad 1 ed effettua il jump a `loc_40103A`



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

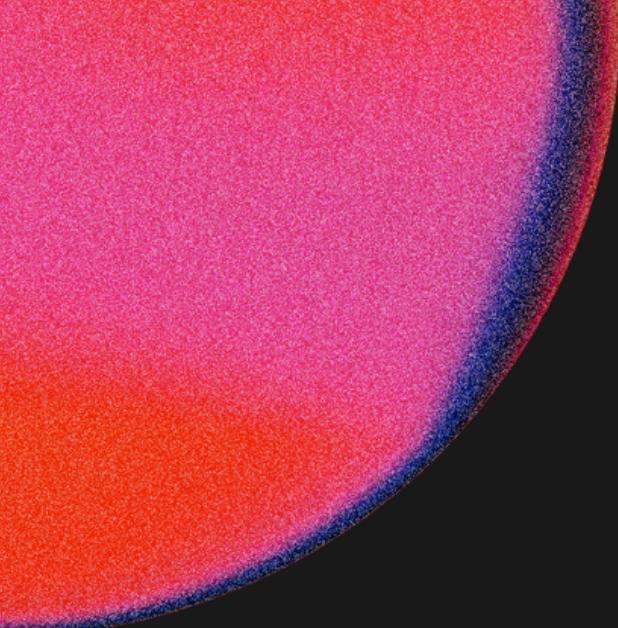
se la condizione precedente risulta essere falsa, chiama la subroutine **sub_40117F** ed esegue l'output della stringa che è stata pushata "**Error 1.1: No Internet\n**", pulisce lo stack e imposta il valore di eax a 0



```
loc_40103A:
mov    esp, ebp
pop    ebp
ret
sub_401000 endp
```

termina la funzione, dopo aver effettuato il **pop** dell' **ebp**. In analisi finale possiamo dire che la funzione implementata serve a verificare se è presente la connessione ad internet o meno.

per l'esecuzione del bonus decido di procedere in prima fase con un analisi statica base, decido di usare quindi md5deep per recuperare l'ash dell'eseguibile ed analizzarlo con il tool VirusTotal



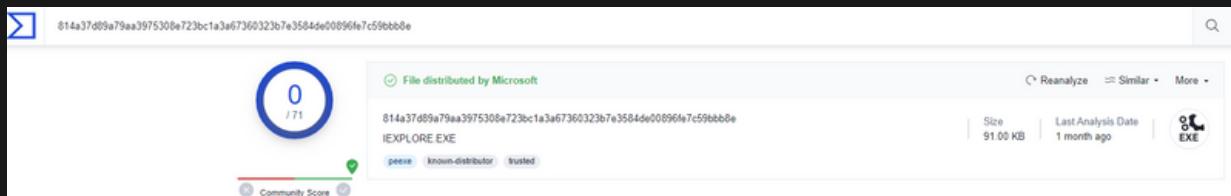
```
Command Prompt
C:\Documents and Settings\Administrator\Desktop>cd md5deep-4.3
C:\Documents and Settings\Administrator\Desktop\md5deep-4.3>dir
Volume in drive C has no label.
Volume Serial Number is D8BA-8021

Directory of C:\Documents and Settings\Administrator\Desktop\md5deep-4.3

08/16/2022  03:37 PM    <DIR>      .
08/16/2022  03:37 PM    <DIR>      -
10/24/2012  02:33 AM           17,715 CHANGES.txt
10/24/2012  02:33 AM           19,422 COPYING.txt
10/24/2012  02:33 AM           2,261 FILEFORMAT.txt
10/24/2012  02:33 AM           800,256 hashdeep.exe
10/24/2012  02:33 AM           12,291 HASHDEEP.txt
10/24/2012  02:33 AM           988,160 hashdeep64.exe
10/24/2012  02:33 AM           800,256 md5deep.exe
10/24/2012  02:33 AM           14,717 MD5DEEP.txt
10/24/2012  02:33 AM           988,160 md5deep64.exe
10/24/2012  02:33 AM           800,256 sha1deep.exe
10/24/2012  02:33 AM           988,160 sha1deep64.exe
10/24/2012  02:33 AM           800,256 sha256deep.exe
10/24/2012  02:33 AM           988,160 sha256deep64.exe
10/24/2012  02:33 AM           800,256 tigerdeep.exe
10/24/2012  02:33 AM           988,160 tigerdeep64.exe
10/24/2012  02:33 AM           800,256 whirlpooldeep.exe
10/24/2012  02:33 AM           988,160 whirlpooldeep64.exe
               17 File(s)   10,796,902 bytes
                2 Dir(s)  52,978,435,536 bytes free
C:\Documents and Settings\Administrator\Desktop\md5deep-4.3>
```

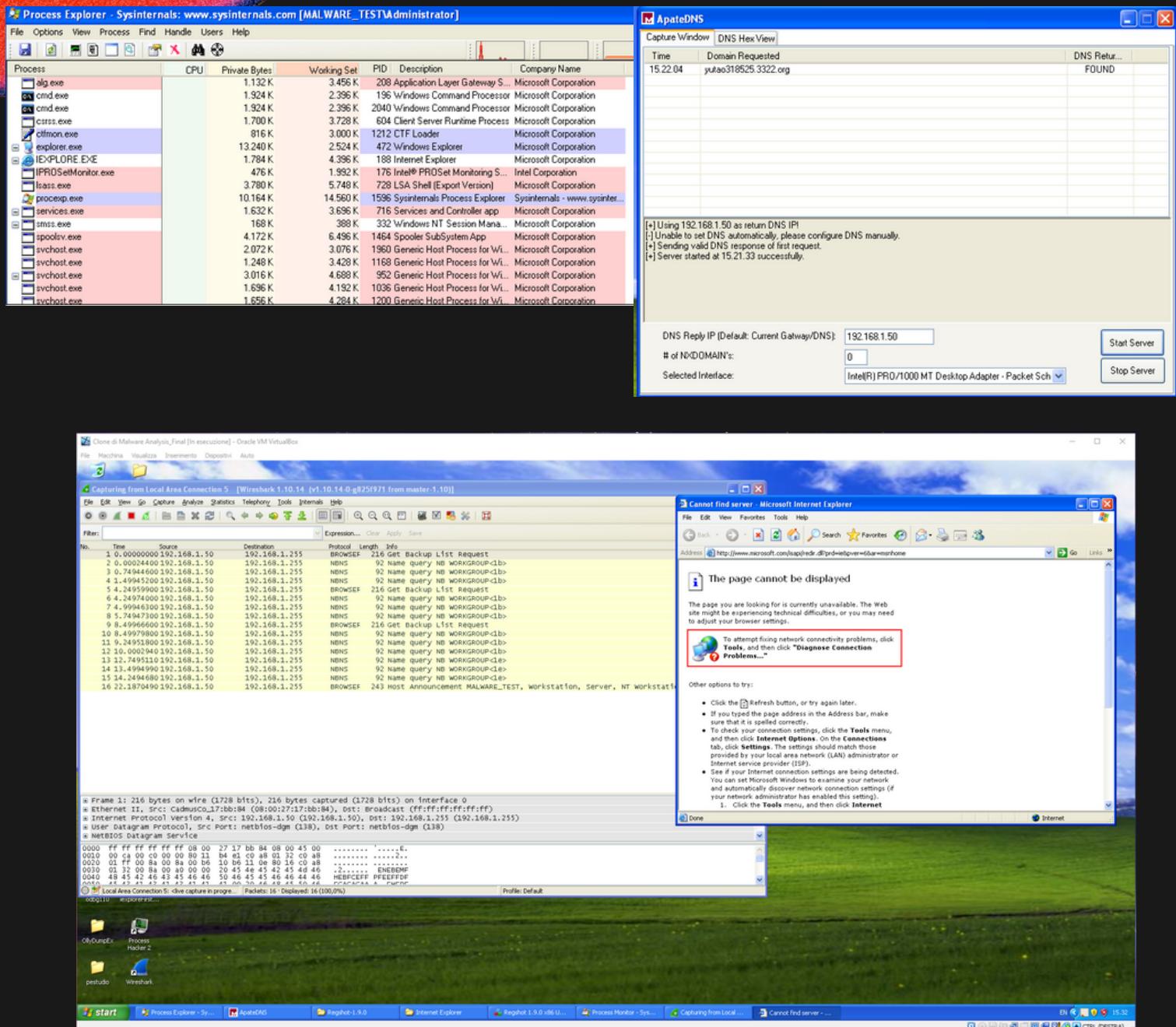
```
C:\Documents and Settings\Administrator\Desktop\md5deep-4.3>md5deep "C:\Program
Files\Internet Explorer\iexplore.exe"
55794b97a7faabd2910873c85274f409  C:\Program Files\Internet Explorer\iexplore.exe

C:\Documents and Settings\Administrator\Desktop\md5deep-4.3>
```



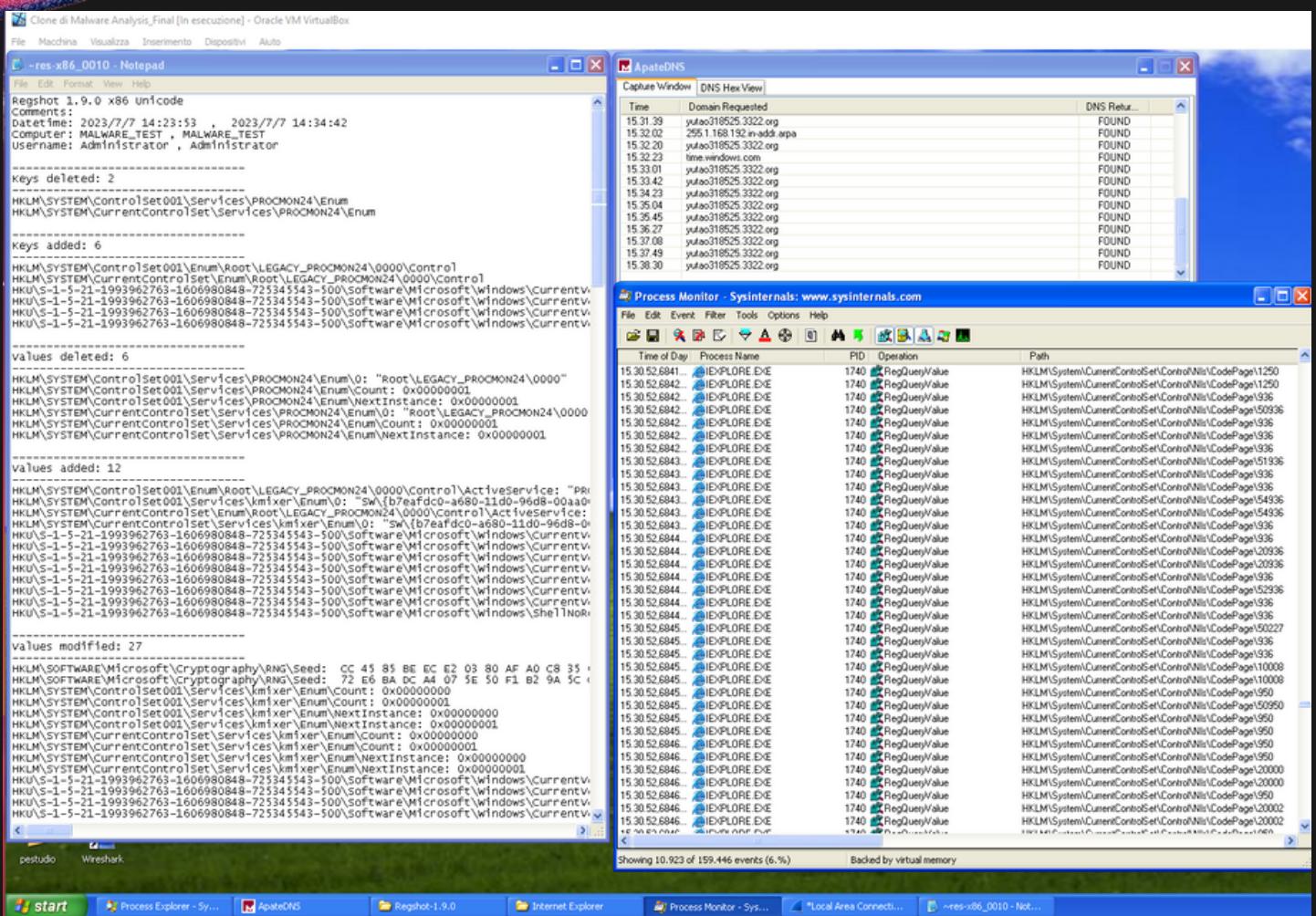
la scansione tramite ash ci riporta ovviamente che l'eseguibile non ha alcun grado di minaccia e che il distributore è Microsoft(quindi noto).

procedo quindi ad un'analisi dinamica base per ulteriore conferma che l'eseguibile non possa arrecare alcun danno alla macchina, prima di effettuare questi test sul pc del collega però scollego il pc da internet (anche se so che il file non è malevolo) perchè non si fida di me (:



avvio process explorer, inizializzo un server dns con apateDNS e salvo una prima istantanea con regshot, successivamente avvio procmon e Wireshark ed eseguo il file, lasciandolo aperto per qualche minuto

dopo aver interrotto la cattura con wireshark e procmon posso registrare una seconda istantanea con regshot e cominciare l'analisi delle modifiche che sono state effettuate dall'eseguibile.



come si evince dallo screen il file effettivamente ha effettuato varie modifiche a chiavi di registro, ha provato a connettersi a internet, ha avviato e terminato vari processi ma dopo un'analisi più approfondita possiamo concludere che nessuna delle modifiche effettuate rientra in una potenziale minaccia per il computer e quindi possiamo tranquillizzare il nostro collega che tornerà a lavoro sereno e contento.