



ESERCITAZIONE

S10 - L5



MICHELANGELO BORROMEO

TRACCIA 1

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:

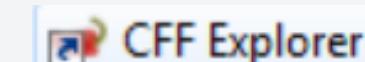
1. Quali **librerie** vengono importate dal file eseguibile? Fare anche una descrizione
2. Quali sono le **sezioni** di cui si compone il file eseguibile del **malware**? Fare anche una descrizione

MALWARE ANALYSIS

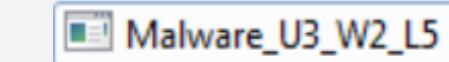
Indicare le librerie importate dal malware, fornendo una descrizione per ognuna di esse

Per andare ad analizzare il Malware in questione, andremo ad utilizzare il tool CFF Explorer
esso ci andrà a fornire relative informazioni riguardo il programma eseguibile.

Dopo aver importato il malware all'interno del tool



open



Possiamo ora andare ad analizzare in profondità l'eseguibile,

nella prima schermata troveremo diverse informazioni:

- Nome
- tipologia di file
- Grandezza file
- data di creazione modifica

Property	Value
File Name	C:\Users\user\Desktop\MALWARE\Esercizio_Pratico_U3_W2_L5\Malw...
File Type	Portable Executable 32
File Info	Microsoft Visual C++ 6.0
File Size	40.00 KB (40960 bytes)
PE Size	40.00 KB (40960 bytes)
Created	Wednesday 02 February 2011, 17.29.06
Modified	Wednesday 17 January 2024, 18.48.15
Accessed	Wednesday 02 February 2011, 17.29.06
MD5	C0B54534E188E1392F28D17FAFF3D454
SHA-1	BB6F01B1FEF74A9CFC83EC2303D14F492A671F3C

MALWARE ANALYSIS

Indicare le librerie importate dal malware, fornendo una descrizione per ognuna di esse

Recandoci sulla sezione



andremo a visualizzare .dll importate da esso.

Malware_U3_W2_L5.exe						
Module Name	Imports	OFTs	TimeStamp	ForwarderChain	Name RVA	FTs (IAT)
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	44	00006518	00000000	00000000	000065EC	00006000
WININET.dll	5	000065CC	00000000	00000000	00006664	000060B4

Come possiamo notare Eseguibile ha importato 2 librerie:

- Kernel32.dll: Fornisce funzioni (44) principali per la gestione della memoria, processi e thread.
- WININET.dll: Fornisce (2)funzioni per l'accesso a Internet e per la gestione delle connessioni HTTP e FTP.

MALWARE ANALYSIS

Indicare le librerie importate dal malware, fornendo una descrizione per ognuna di esse

Può essere utilizzato per:

Kernel32.dll:

- Creare processi e tread
- Manipolare la memoria
- Avere accesso ai file di sistema
- E reperire le caratteristiche (OS)

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
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

LoadLibraryA

carica una libreria dinamica (DLL) nel processo chiamante.
Il processo può chiamare le funzioni esportate dalla DLL.

GetProcAddress

Recupera l'indirizzo di una funzione esportata dalla DLL caricata.

Può essere utilizzato per:

WININET.dll:

- Effettuare download di payload aggiuntivi
- Connetersi a server C2 (command and control)

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

InternetOpenA

utilizzata per configurare la connessione a Internet e ottenere un handle per ulteriori operazioni di rete, come l'apertura di URL, l'invio di richieste HTTP, ecc.

MALWARE ANALYSIS

Indicare le librerie importate dal malware, fornendo una descrizione per ognuna di esse

Per andare a vedere le sezioni dell'eseguibile ci rechiamo su

Section Headers [x]

Malware_U3_W2_L5.exe										
Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N...	Linenumbers ...	Characteristics	
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword	
.text	00004A78	00001000	00005000	00001000	00000000	00000000	0000	0000	60000020	
.rdata	0000095E	00006000	00001000	00006000	00000000	00000000	0000	0000	40000040	
.data	00003F08	00007000	00003000	00007000	00000000	00000000	0000	0000	C0000040	

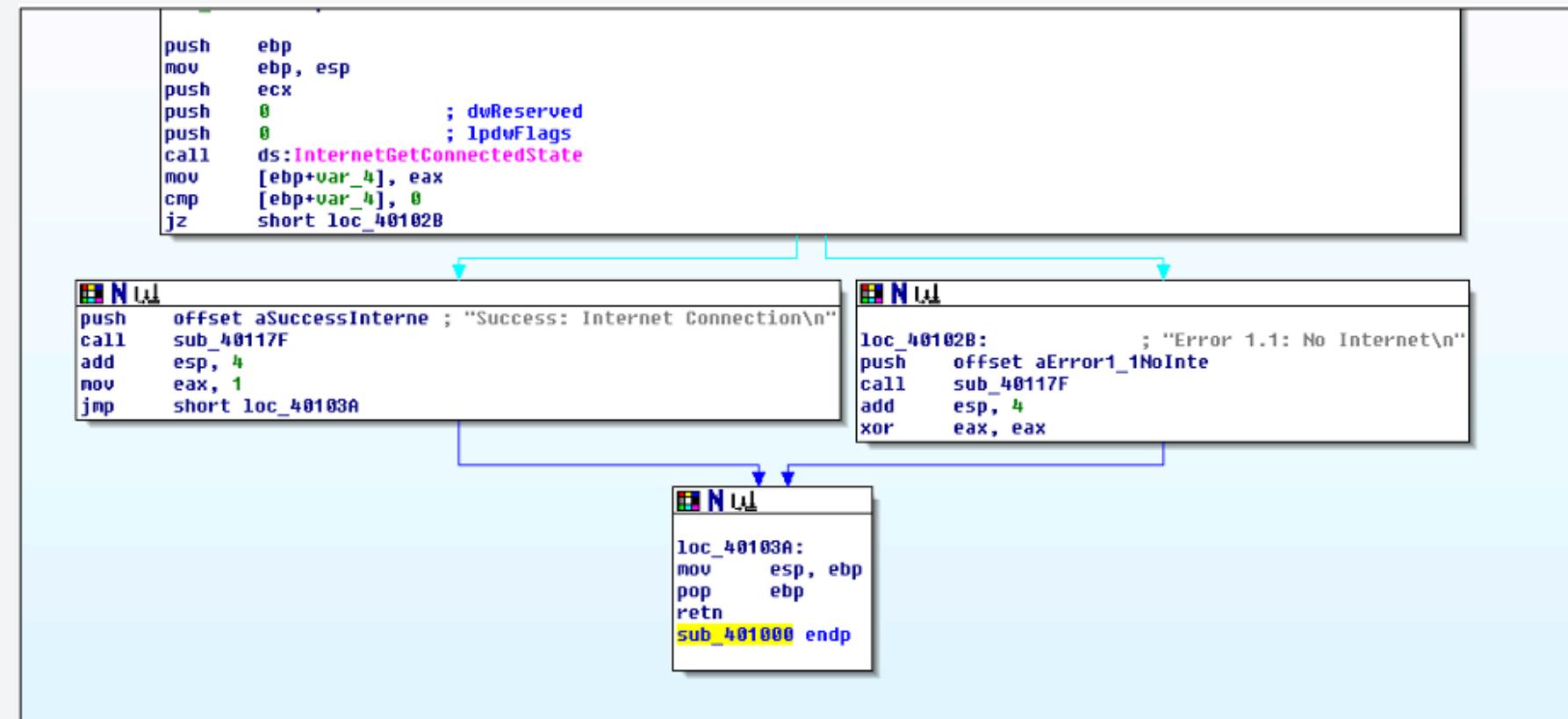
In particolare noteremo 3 sezioni:

- .text
- .rdata
- .data

TRACCIA 2

Con riferimento alla figura in slide 3, risponde ai seguenti quesiti:

3. Identificare i **costrutti** noti (creazione dello stack, eventuali cicli, altri costrutti)
4. Ipotizzare il comportamento della funzionalità implementata
5. Fare una tabella per spiegare il significato delle singole righe di codice



CASO 1

Internet Connection
success

The screenshot shows a debugger interface with three windows displaying assembly code:

- Top Window:** Shows the initial assembly code. A blue arrow points from the `jz` instruction at address `loc_40102B` to the `loc_40103A` label in the middle window.
- Middle Window:** Shows the assembly code for the `loc_40103A` label, which contains the message "Success: Internet Connection\n".
- Bottom Window:** Shows the assembly code for the `sub_401000` procedure, which includes the `ret` instruction.

```
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

; Success: Internet Connection\n

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

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
```

CREAZIONE STACK

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

push EBP

Inserisce EBP nello stack

```
mov    EBP,ESP
```

Copia il valore di ESP all'interno di EBP

CREAZIONE PARAMETRI E CHIAMATA FUNZIONE

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

push 0 ; dwReserved

Inserisce il valore 0 nello stack con commento ; dwReserved

push 0 ; lpdwFlags

Inserisce 0 nello stack con commento ; lpdwFlags

call ds:InternetGetConnectedState

Chiama la funzione con i parametri: dwReserved - lpdwFlags

CLICLO ' IF '

```
cmp    [ebp+var_4], 0  
jz     short loc_40102B
```

cmp [EBP+var_4], 0

Effettua una comparazione tra 0 e il valore di [EBP+var_4]

jz short loc_40102B.

Se il risultato del confronto è zero, salta all'etichetta loc_40102B

ASSEMBLY

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

CREAZIONE PARAMETRI E CHIAMATA FUNZIONE

push offset aSuccessInterne
"Success: Internet Connection\n"

call sub_40117F

jmp short loc_40103A

push offset aSuccessInterne
Inserisce l'offset all'interno
della stringa

call sub_40117F
Richiama la funzione sub_40117F

jmp short loc_40103A.
Se il risultato del confronto è zero, salta all'etichetta loc_40102A

ASSEMBLY

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

Fine Funzione

mov esp, ebp

mov esp, ebp

Copia ebp all'interno di esp

pop ebp

pop ebp

Rimuove l'ultimo valore salvato
nello stack e lo carica nel
registro ebp

ret

ret

ritorno della funzione

sub_401000 endp

sub_401000 endp

Fine definizione della
funzione.

CASO 2

Internet Connection failed

The screenshot shows a debugger interface with two assembly code snippets. The top window, titled 'NUL', contains the following code:

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

The bottom window, also titled 'NUL', contains the following code:

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

A blue arrow points from the end of the first window's code to the start of the second window's code, indicating a flow or continuation between them.

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
```

CREAZIONE STACK

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

push EBP

Inserisce EBP nello stack

```
mov    EBP,ESP
```

Copia il valore di ESP all'interno di EBP

CREAZIONE PARAMETRI E CHIAMATA FUNZIONE

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

push 0 ; dwReserved

Inserisce il valore 0 nello stack con commento ; dwReserved

push 0 ; lpdwFlags

Inserisce 0 nello stack con commento ; lpdwFlags

call ds:InternetGetConnectedState

Chiama la funzione con i parametri: dwReserved - lpdwFlags

CLICLO ' IF '

```
cmp    [ebp+var_4], 0  
jz     short loc_40102B
```

cmp [EBP+var_4], 0

Effettua una comparazione tra 0 e il valore di [EBP+var_4]

jz short loc_40102B.

Se il risultato del confronto è zero, salta all'etichetta loc_40102B

ASSEMBLY

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

CHIAMATA FUNZIONE con PARAMETRI

push offset aSuccessInterne
"Success: Internet Connection\n"

call sub_40117F

xor eax, eax

push offset AError1_1NoInte

Inserisce l'offset all'interno di
errore dello stack

call sub_40117F

Richiama la funzione sub_40117F

xor eax,eax

Imposta il registro EAX a 0 (indicando il fallimento).

ASSEMBLY

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

FINE FUNZIONE

mov esp, ebp

mov esp, ebp

Copia ebp all'interno di esp

pop ebp

pop ebp

Rimuove l'ultimo valore salvato
nello stack e lo carica nel
registro ebp

ret

ret

ritorno della funzione

sub_401000 endp

sub_401000 endp

Fine definizione della
funzione.

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

loc_40102B:           ; "Success: Internet Connection\n"
push    offset aSuccessInterne ; "Success: Internet Connection\n"
call    sub_40117F
add    esp, 4
mov    eax, 1
jmp    short loc_40103A

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

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

Il codice ha la funzione di verificare la connessione ad internet:

- se la connessione avviene, mostra un messaggio di successo
- se la connessione non avviene, ci mostra un messaggio di errore
- Infine ripristina lo stack e termina la funzione

BONUS

BONUS:

Un giovane dipendente neo assunto segnala al reparto tecnico la presenza di un programma sospetto.

Il suo superiore gli dice di stare tranquillo ma lui non è soddisfatto e chiede supporto al SOC.

Il file "sospetto" è `iexplore.exe` contenuto nella cartella C :\Programmi \Internet Explorer
(no, non ridete ragazzi)

Come membro senior del SOC ti è richiesto di convincere il dipendente che il file non è maligno.

Possono essere usati gli strumenti di analisi statica basica e/o analisi dinamica basica visti a lezione.
No disassembly no debug o similari

VirusTotal non basta, ovviamente

Non basta dire iexplorer è Microsoft quindi è buono, punto.

ANALISI STATICÀ BASICA

Possiamo andare ad utilizzare CFF Explore per andare a recuperare informazioni riguardo l'integrità dell'applicativo.

Property	Value
File Name	C:\Program Files (x86)\Internet Explorer\iexplore.exe
File Type	Portable Executable 32
File Info	No match found.
File Size	657.27 KB (673040 bytes)
PE Size	650.50 KB (666112 bytes)
Created	Sunday 21 November 2010, 05.25.08
Modified	Sunday 21 November 2010, 05.25.08
Accessed	Sunday 21 November 2010, 05.25.08
MD5	C613E69C3B191BB02C7A191741A1D024
SHA-1	1962888198AE972CBB999D0DC9C9EE5CBABF5E0D

Importando iexplore.exe avremo:

- nome
- dimensione
- date creazione/ultima modifica
- Hash MD5
- Sha -1

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word
.text	0000A085	00001000	0000A200	00000400	00000000	00000000	0000
.data	00000618	0000C000	00000600	0000A600	00000000	00000000	0000
.rsrc	00097020	0000D000	00097200	0000AC00	00000000	00000000	0000
.reloc	00000B8C	000A5000	00000C00	000A1E00	00000000	00000000	0000

Nella sezione

Section Headers [x]

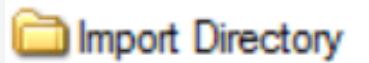
noteremo la presenza di 4

- .text
- .data
- .rsrc
- .reloc

Solitamente nei Malware le sezioni vengono in qualche modo nascoste con parole sospette

ANALISI STATICÀ BASICA

Recandoci sulla sezione



andremo a visualizzare .dll importate da esso.

Module Name	Imports	OFTs	TimeStamp	ForwarderChain	Name RVA	FTs (IAT)
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
ADVAPI32.dll	9	0000A518	FFFFFFF	FFFFFFF	0000A508	00001000
KERNEL32.dll	59	0000A540	FFFFFFF	FFFFFFF	0000A4F8	00001028
USER32.dll	9	0000A630	FFFFFFF	FFFFFFF	0000A4EC	00001118
msvcrt.dll	28	0000A658	FFFFFFF	FFFFFFF	0000A4E0	00001140
ntdll.dll	1	0000A6CC	FFFFFFF	FFFFFFF	0000A4D4	000011B4
SHLWAPI.dll	18	0000A6D4	FFFFFFF	FFFFFFF	0000A4C8	000011BC
SHELL32.dll	2	0000A720	FFFFFFF	FFFFFFF	0000A4BC	00001208
ole32.dll	2	0000A72C	FFFFFFF	FFFFFFF	0000A4B0	00001214
iertutil.dll	14	0000A738	FFFFFFF	FFFFFFF	0000A4A0	00001220
urlmon.dll	3	0000A774	FFFFFFF	FFFFFFF	0000A494	0000125C

Come possiamo notare, a differenza dei malware, abbiamo tante librerie con diverse funzioni

Solitamente nei Malware sono presenti solamente 2 librerie con le funzioni principali come
GetProcAddress
LoadLibraryA

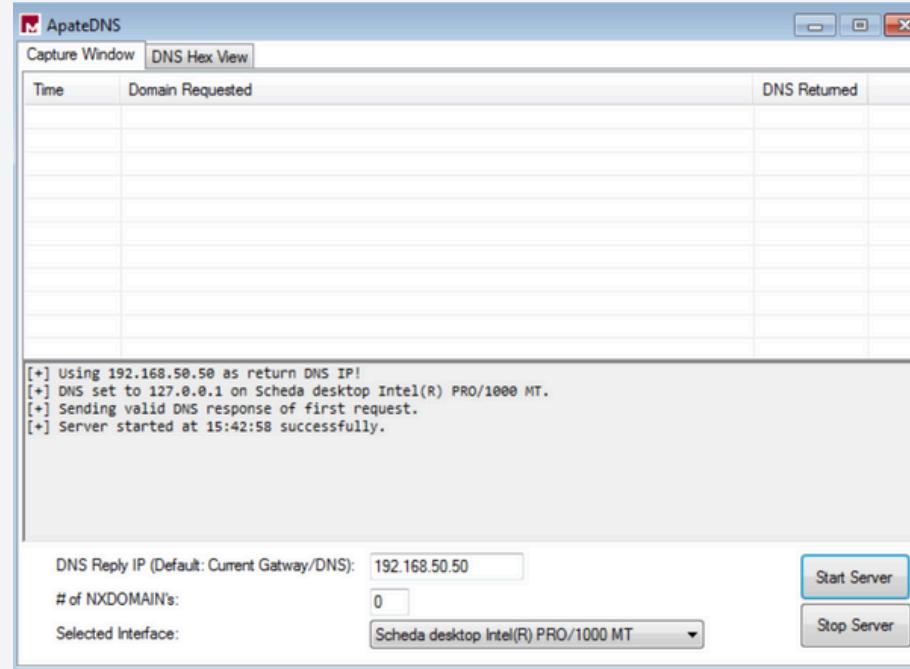
ANALISI STATICÀ BASICA

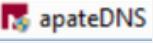
The screenshot shows the VirusTotal analysis interface for the file iexplore.exe. The file has an MD5 hash of e285feeca968b3ca22017a64363eea5e69ccd519696671df523291b089597875. The analysis was performed 2 months ago, with a size of 657.27 KB. The interface displays a community score of 0 / 73. Below the main header, there are tabs for DETECTION, DETAILS, RELATIONS, BEHAVIOR, and COMMUNITY (with 30+ entries). A call-to-action button 'Join our Community' is present. Under 'Security vendors' analysis', results for Acronis (Static ML), AhnLab-V3, Alibaba, AliCloud, and ALYac are shown, all marked as 'Undetected'. A section for 'Do you want to automate checks?' is also visible.

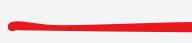
Analizziamo l'Hash MD5 preso da CFF Explore, su diversi Tool di verifica come virus total, Metadefender Cloud, kaspersky

The screenshot shows the OPSWAT MetaDefender Cloud analysis interface for the file iexplore.exe. The file hash is C613E69C3B191BB02C7A191741A1D024. The analysis is labeled as 'Overview'. It states 'No threats found on this file.' and includes a note: 'A more in-depth analysis can be performed using our Sandbox technology. Click the 🔄 button and resubmit the file with the Sandbox option checked to get the dynamic analysis results.' Below this, it says 'The file is not sanitizable'. The interface includes sections for 'Metascan' (0 threats detected, 0 engines), 'Sandbox Score' (0 dynamic analysis performed, 0%, 0 votes), and 'Community Insight' (0%, 0 votes). Buttons for 'Get full report', 'View summary', 'Upgrade limits', 'Sandbox documentation', and 'View leaderboards' are also present.

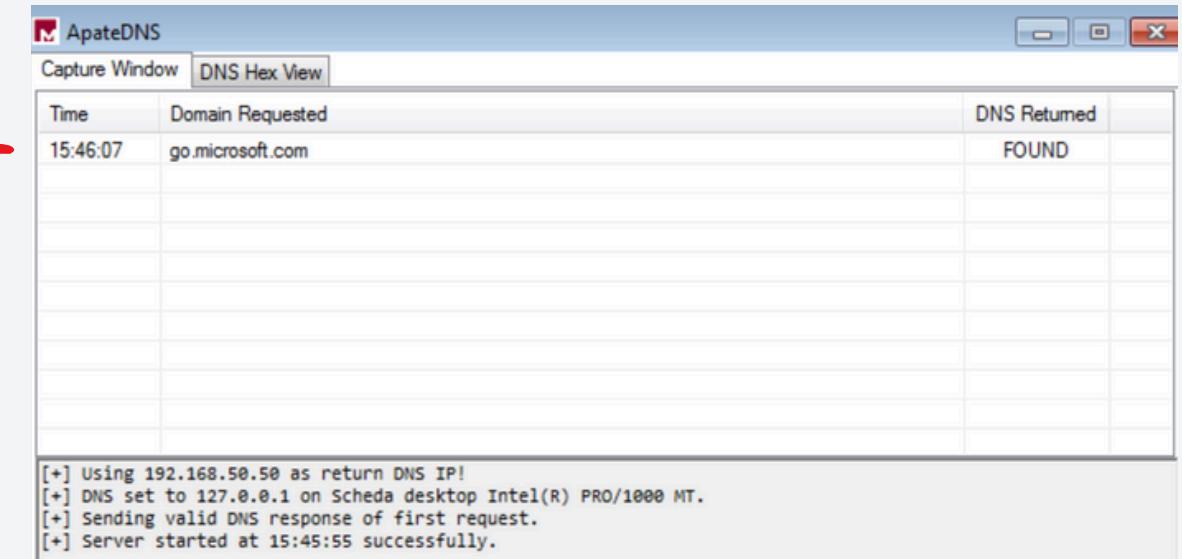
ANALISI DINAMICA BASICA



Utilizziamo anche  per monitorare e reindirizzare le richieste DNS fatte dal software in questione, è utile per analizzare il comportamento del malware e capire quali domini cerca di contattare.

go.microsoft.com 

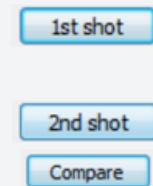
In questo caso possiamo notare che l'eseguibile non ci reindirizza su siti sospetti.



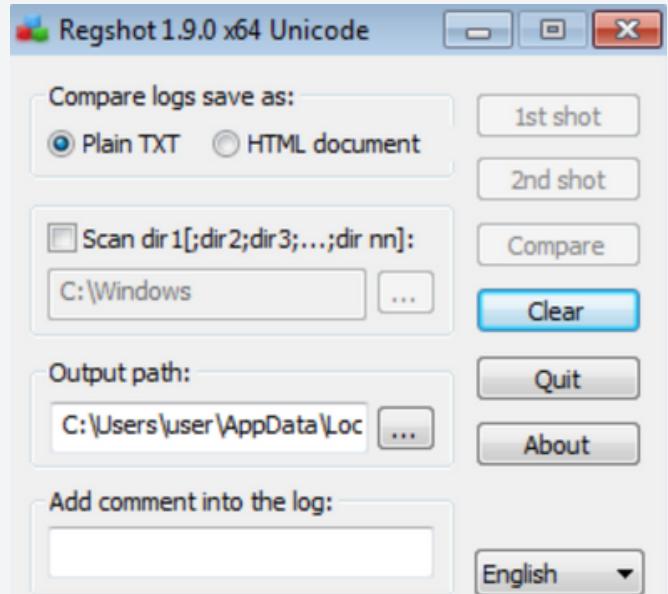
ANALISI DINAMICA BASICA

Con l'utilizzo di  possiamo andare a confrontare i valori aggiunti o modificati prima e dopo aver eseguito il software in questione

- 1) Si effettuerà una prima scansione
 - 2) Eseguiamo il Software
 - 3) si eseguirà una seconda scansione
 - 4) clicchiamo su Compare



Dopo alcuni secondi avremo un file txt nella quale Regshot andrà a segnare tutti i valori modificati o aggiunti, in questo caso non verranno aggiunti o eliminati valori ma modificati.



- Valori aggiunti: 0
 - Valori modificati: 3
 - Valori eliminati: 0

SOLUZIONE

In base ai vari test eseguiti:

- Analisi statica basica

Utilizzo di software come CFF Explore, VirusTotal e MetaDefender.

- Analisi dinamica basica

Utilizzo di software come ApateDNS e Regshot

Possiamo dedurre che L'eseguibile è Innocuo e può essere eseguito.