

S10L5

Rosario Z.

Indice

Traccia.....	pag 1
Concetti chiave.....	pag 2
Traccia 1 Punto 1.....	pag 4
Traccia 1 Punto 2.....	pag 5
Traccia 2 Punto 1.....	pag 6
Traccia 2 Punto 2.....	pag 8
Approfondimento Traccia 1.....	pag 9
Conclusioni.....	pag 11

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:

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

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

- Identificare i costrutti noti (creazione dello stack, eventuali cicli, costrutti)
- Ipotezzare il comportamento della funzionalità implementata

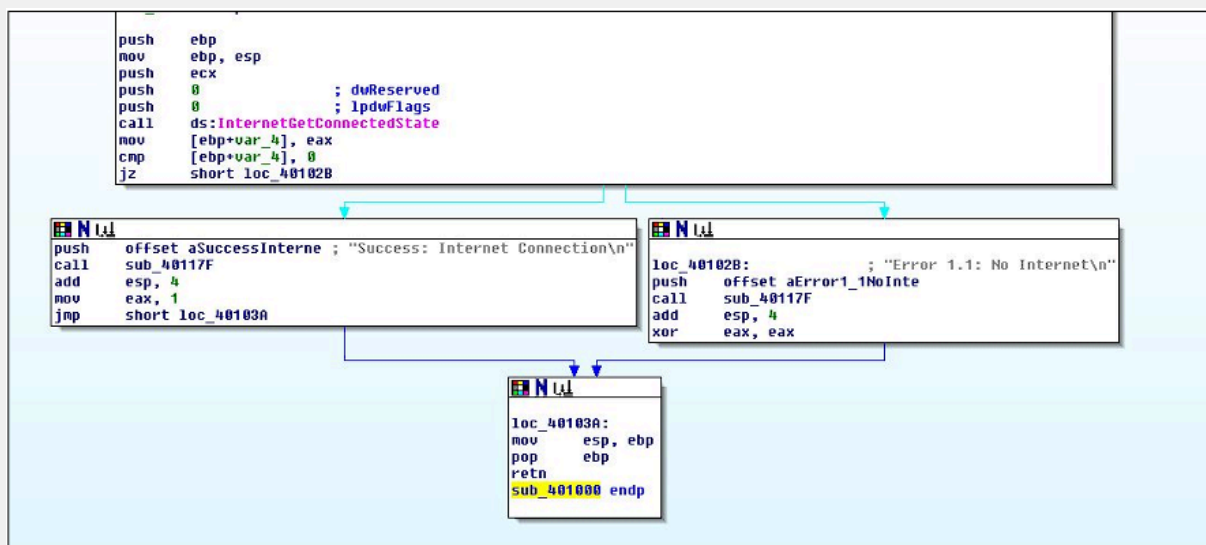


Figura in slide 3

Concetti chiave:

Sarebbe bene anticipare subito alcuni concetti fondamentali per capire di cosa parleremo successivamente affrontando i punti proposti dalla traccia.

L'analisi sarà effettuata su un file .EXE, cioè un file eseguibile.
All'interno di questo file siamo sicuri di trovare un Malware.

Malware:

Un malware (software malevolo) è un programma informatico progettato per danneggiare un sistema informatico o per ottenere un accesso non autorizzato a dati e risorse dello stesso dispositivo.

I malware possono assumere diverse forme, tra le quali troviamo:

- **Virus:** si replicano e si diffondono da un computer all'altro, infettando i file e i sistemi
- **Worm:** si diffondono attraverso le reti informatiche, sfruttando le vulnerabilità del sistema
- **Trojan horse:** si nascondono all'interno di altri programmi o file, ingannando l'utente a installarli
- **Spyware:** raccolgono informazioni personali e sensibili dell'utente senza il suo consenso
- **Ransomware:** criptano i file dell'utente e chiedono un riscatto per decriptare i file

La forma definisce anche le azioni che compie.

Le librerie che vengono importate da un file possono farci distinguere un eseguibile pulito da uno infetto.

Le sezioni di intestazione di cui si compone un file può darci indizi su come è stato strutturato quel determinato file.

I Software che andremo ad utilizzare servono ad analizzare i file eseguibili che abbiamo il sospetto che contengono Malware.

Uno dei Software utili a questo fine che andremo ad utilizzare si chiama CFF Explorer VIII.

Assembly:

Il linguaggio assembly è un linguaggio di programmazione a basso livello. Questo linguaggio ci permette di leggere comprensibilmente le azioni eseguite dalla macchina che andrebbero altrimenti lette in codice macchina. Ogni istruzione assembly corrisponde a un'operazione effettuata dal processore, come ad esempio:

- Spostamento dei dati all'interno dei registri
- Operazioni aritmetiche che portano al risultato di un processo
- Controllo del flusso di dati

Vantaggi:

In breve: Fra i vantaggi troviamo il fatto che l'assembly è un linguaggio assai vicino al linguaggio macchina e dunque assai vicino all'hardware.

Svantaggi:

In breve: Fra gli svantaggi troviamo sicuramente la scarsa facilità di lettura del codice se scritto in Assembly.

Utilizzo:

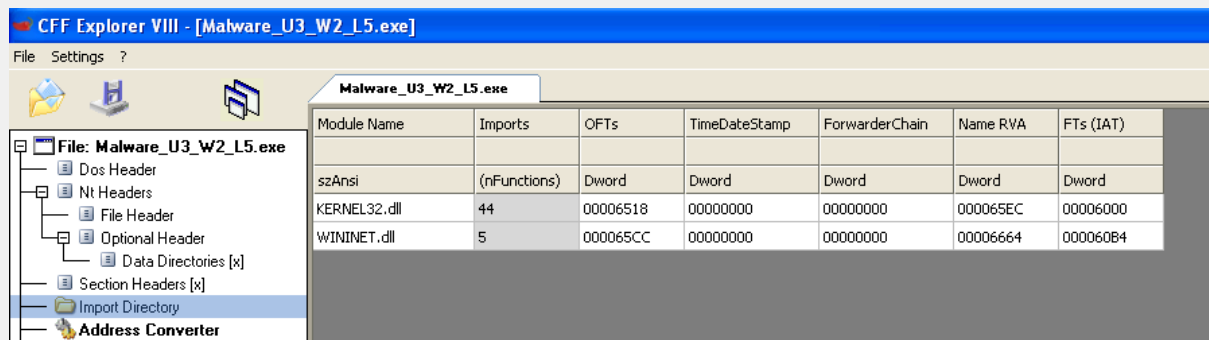
Utilizzeremo Assembly per leggere i file eseguibili e dedurre il loro comportamento.

Traccia 1

Punto 1

- Quali librerie vengono importate dal file eseguibile?

Per analizzare il file al fine di capire quali librerie vengono importate dal file malevolo utilizziamo il Software “CFF Explorer VIII”.



CFF Explorer VIII - [Malware_U3_W2_L5.exe]

File Settings ?

Malware_U3_W2_L5.exe

Module Name	Imports	OFTs	TimeDateStamp	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

Figura 1

Come da Figura 1, una volta eseguito CFF Explorer e caricato il nostro eseguibile ci siamo spostati nella sezione dedicata alle cartelle importate, che si chiama “Import Directory”.

Da questa sezione è possibile osservare come il File eseguibile stia effettivamente utilizzando delle Directory sensibili, va infatti ad importare:

- KERNEL32.dll
- WININET.dll

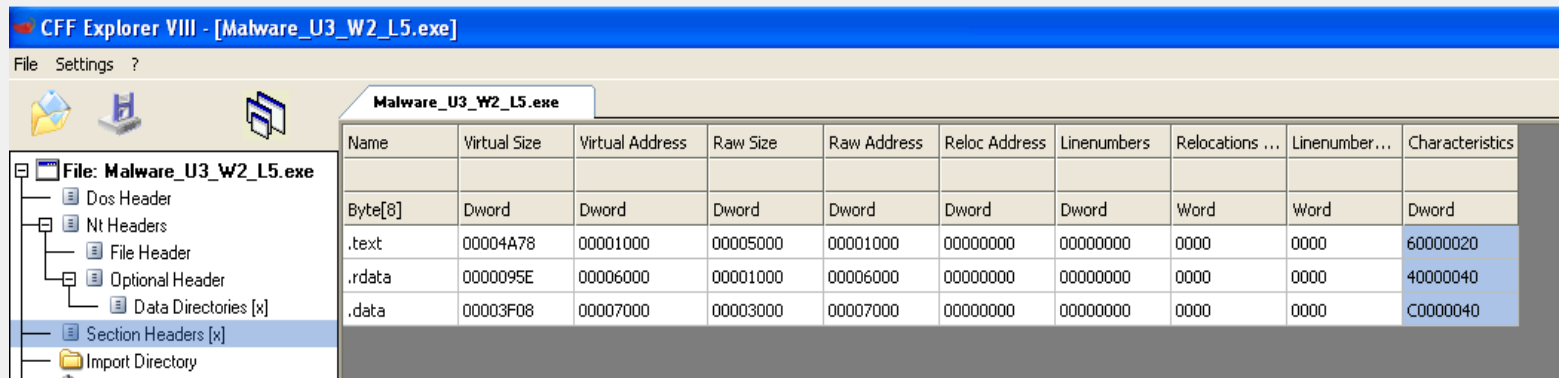
Entrambe le Directory non avrebbero motivo di essere importate all'interno dell'esecuzione di un software se non per scopi malevoli.

Possiamo già ipotizzare si tratti di un Malware, alcune informazioni aggiunte potrebbero aiutarci a dedurre il tipo di malware in esecuzione o comunque le azioni che compie.

Traccia 1

Punto 2

- Quali sono le sezioni di cui si compone il file eseguibile del malware?



Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations ...	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

Figura 2

Utilizzando CFF Explore ci andiamo a dirigere all'interno della posizione "Section Headers".

All'interno di questa posizione troveremo le sezioni di cui si compone il file eseguibile.

In questo caso possiamo notare dalla Figura 2 come il Malware sia diviso in tre sezioni, che sono le seguenti:

- .text
- .rdata
- .data

Traccia 2

Punto 1

- **Identificare i costrutti noti (creazione dello stack, eventuali cicli, costrutti)**

Ciò che ci troviamo davanti in Figura 3 è lo schema di un codice Assembly. Tra i costrutti noti, troviamo:

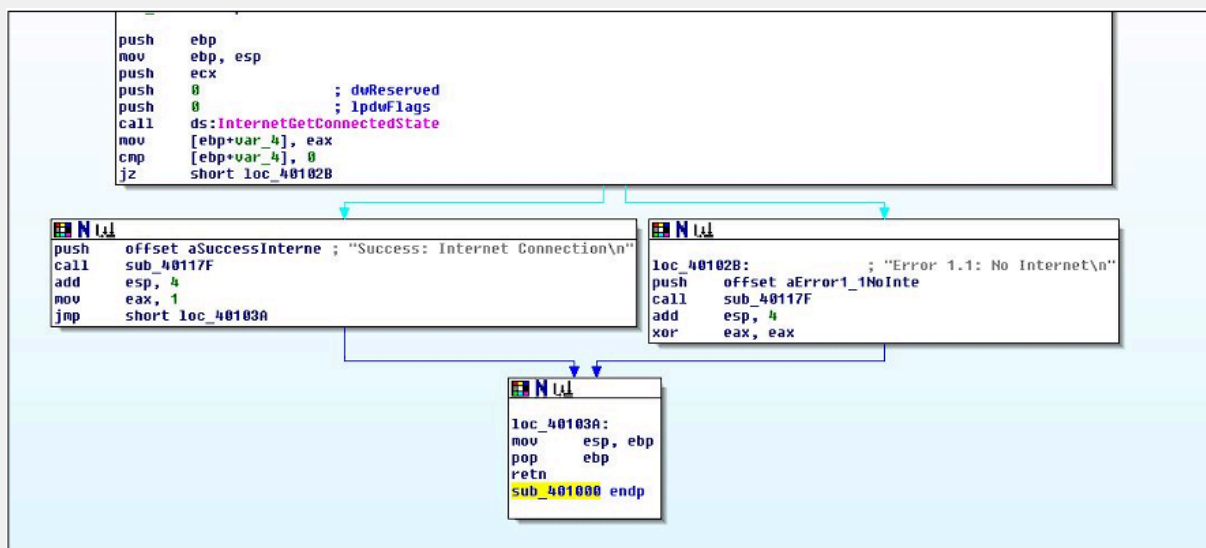


Figura 3

Creazione dello stack tramite il blocco di istruzioni:

Push ebp: Salva il valore del registro ebp sullo stack.

Mov ebp, esp: Imposta il registro ebp su esp

Push ecx: Salva il valore del registro ecx sullo stack.

Push 1pdwFlags: Salva il valore della variabile 1pdwFlags sullo stack.

Call ds:InternetGetConnectedState: Chiama la funzione `InternetGetConnectedState`.

Mov [ebp+var_4], eax: Memorizza il valore restituito dalla funzione `InternetGetConnectedState` nella variabile `[ebp+var_4]`.

Costrutto condizionale «IF», identificato dalla coppia di istruzioni:

Cmp [ebp+var_4], 8: Confronta il valore della variabile [ebp+var_4] con 8.
Jz short loc_401028: Se il valore della variabile [ebp+var_4] è uguale a 8, salta alla posizione loc_401028.

Push offset aSuccessInterne: Salva l'indirizzo della stringa "Success: Internet Connection\n" sullo stack.

Call sub_40117F: Chiama la funzione sub_40117F per stampare la stringa "Success: Internet Connection\n".

Jmp short loc_40103A: Salta alla posizione loc_40103A.

Loc_401028: Push offset aError1: Salva l'indirizzo della stringa "Error 1.1: No Internet\n" sullo stack.

Add esp, 4: Incrementa il valore del registro esp di 4.

Push 1NoInte: Salva il valore 1 sullo stack.

Call sub_40117F: Chiama la funzione sub_40117F per stampare la stringa "Error 1.1: No Internet\n".

Add esp, 4: Incrementa il valore del registro esp di 4.

Jmp short loc_40103A: Salta alla posizione loc_40103A.

Rimozione dello stack tramite le istruzioni:

Mov esp, ebp: Ripristina il valore del registro esp da ebp

Pop ebp: Ripristina il valore del registro ebp dallo stack

Retn: Restituisce il controllo al chiamante

Traccia 2

Punto 2

- **Ipotizzare il comportamento della funzionalità implementata**

Abbiamo trovato la funzione `getinternetconnectstate`, che viene utilizzata per controllare se su una macchina è presente o meno la connessione ad internet. Questo codice serve a restituire in risposta la presenza o assenza della connessione ad internet.

Il costrutto IF «controlla» se il parametro restituito dalla funzione `getinternetconnectstate` è uguale a 0.

- Nel caso in cui fosse uguale a 0, la funzione stampa a schermo la scritta “No internet” e completa successivamente l’esecuzione del resto del codice.
- Se invece il valore di ritorno della funzione “`getinternetconnectstate`” è diverso da 0, allora la funzione stampa a schermo la scritta “Success: Internet Connection”, successivamente continua l’esecuzione del codice fino alla sua fine.

Il codice completo del Malware potrebbe tuttavia essere diverso da quello presentato.

Potrebbe infatti essere compito del malware assicurarsi della connessione ad internet inizialmente per poi scaricare file malevoli in maniera nascosta alla vista e alla conoscenza della vittima e della sua macchina.

Potrebbe trattarsi di un Downloader, cioè di un Malware dedito a scaricare da Internet ulteriori Malware.

Approfondimento Traccia 1:

Per approfondire possiamo andare a vedere come si comporta un altro programma utilizzato sempre nel campo della Malware Analysis. Nelle Figura 4-5-6, andremo a vedere l'utilizzo del software "ExeInfo". Una volta avviato il Software andremo a caricare il File eseguibile all'interno del nostro Tool.

In Figura 4 Vediamo la schermata che visualizzeremo una volta caricato il malware all'interno del nostro tool.

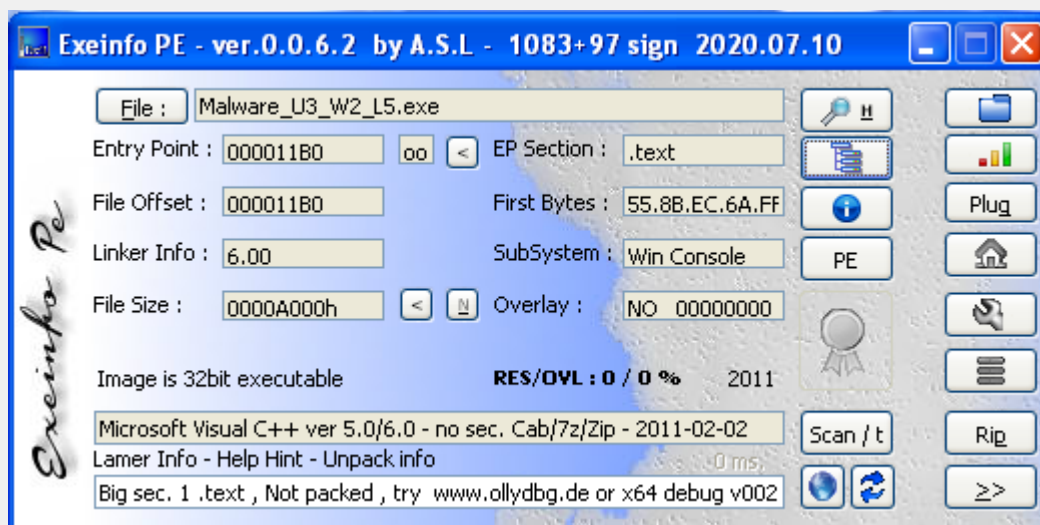


Figura 4

Come evidenziato dalla Figura 5 e dalla Figura 6 ciò che possiamo osservare è una situazione simile a quella di cui avevamo parlato già. Utilizzando questo Tool riusciamo dunque a capire quali funzioni svolge il nostro Malware di riferimento.

Pensiamo che lo schema riportato per lo svolgimento della Traccia 2 sia lo stesso che potremmo ottenere analizzando il malware della Traccia 1, in quanto:

Anche in questo caso infatti vediamo delle istruzioni che sembrano voler ricavare informazioni sullo stato di connessione ad internet della macchina infetta.

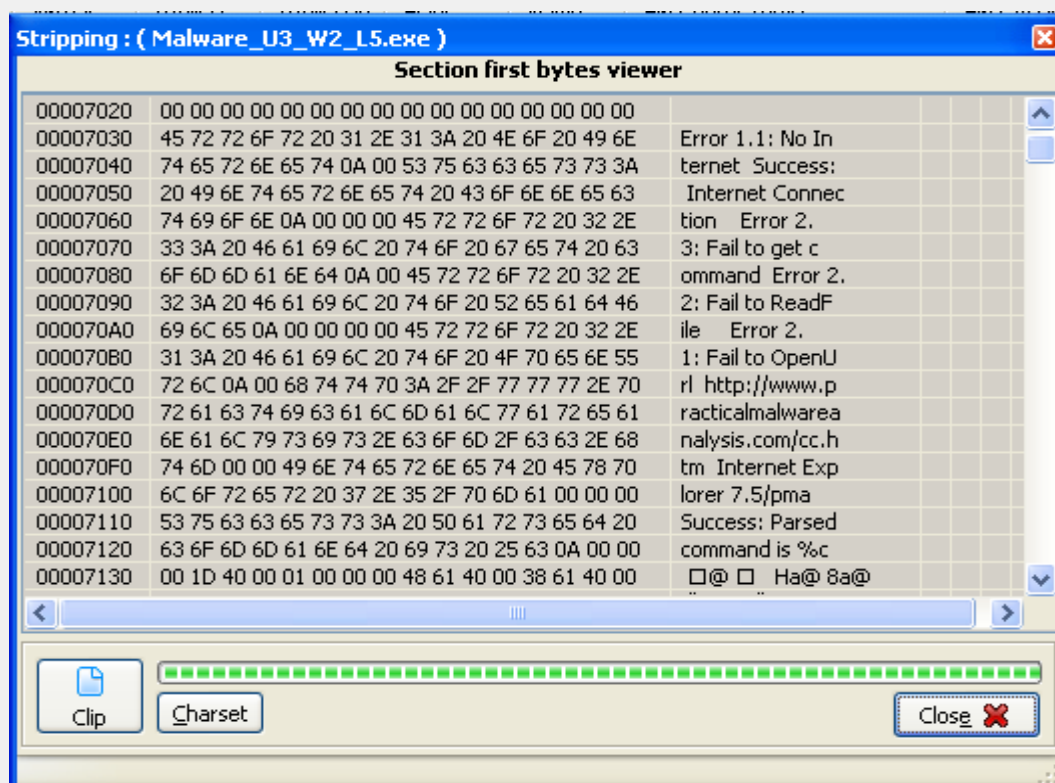


Figura 5

Inoltre come visibile dalla Figura 6, il nostro Malware va a richiamare altre funzioni utili all'individuazione di altre importanti informazioni.

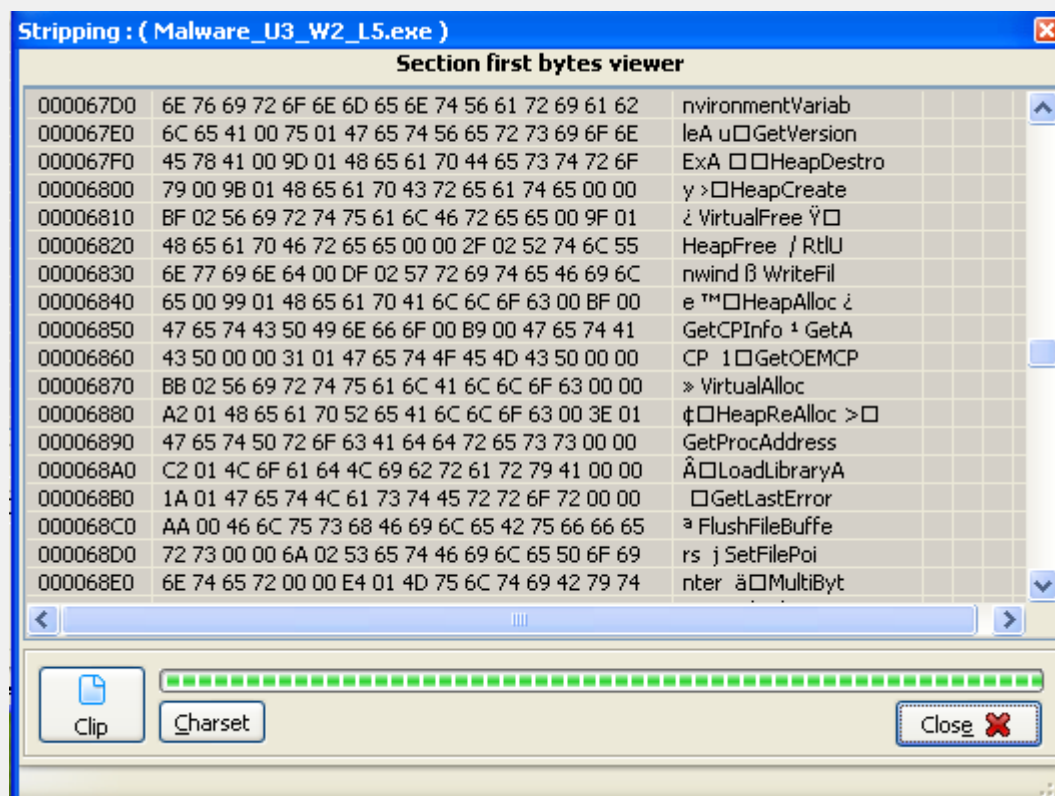


Figura 6

Conclusioni:

Come già detto il nostro Malware sembra compiere soltanto un'azione, cioè assicurarsi della connessione ad internet della macchina vittima.

Come già sottolineato potrebbe però trattarsi in realtà di un Downloader, di conseguenza quella di accertarsi della connessione ad Internet potrebbe essere null'altro che un'operazione preliminare.

Vanta_Black