

Modulo 06

Candidato Angelo Marino

SOMMARIO

- Analisi Statica
- Parametri e variabili
- Identificazioni sezioni
- Librerie malware
- Ipotesi Malware
- Funzioni malware
- Analisi dinamica
- Malware analysis
- Conclusioni

Cos'è l'analisi Statica ?

L'analisi statica del malware è un metodo che consiste nell'esaminare il codice del malware senza eseguirlo. Gli analisti cercano di identificare le funzionalità dannose, estrarre informazioni come indirizzi IP o URL, e creare firme per il rilevamento. Questo processo aiuta a comprendere il comportamento del malware senza doverlo eseguire direttamente.

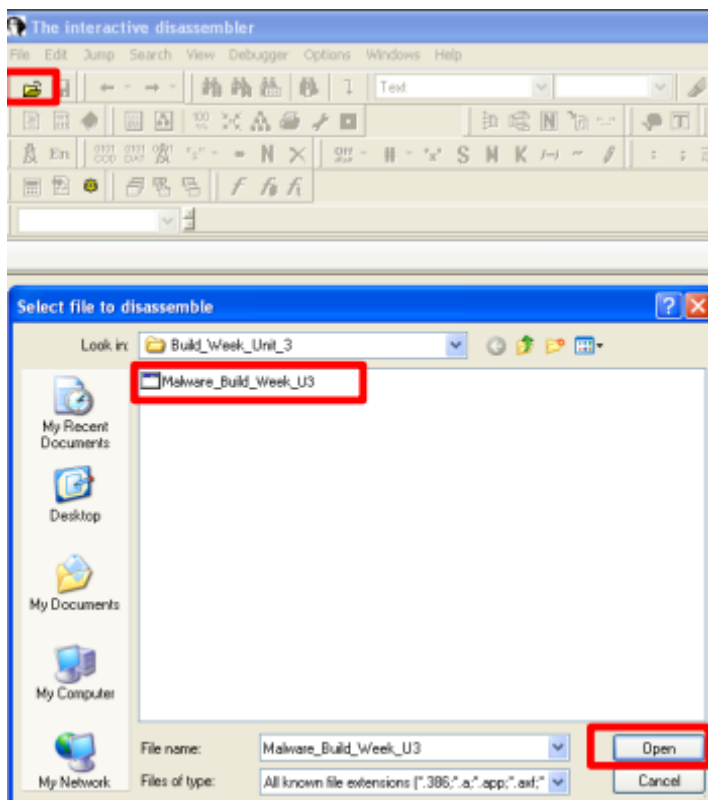
Per L'analisi del Malware_Build_Week_U3 utilizzeremo IDA ricordando di eseguire tutto ciò in un ambiente sicuro

IDA

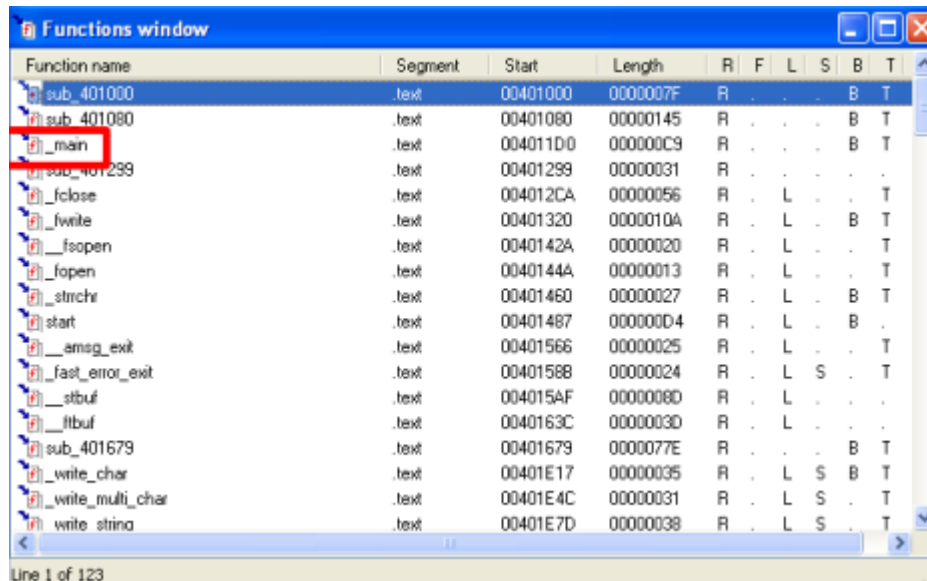
IDA Pro è uno strumento informatico usato per analizzare il codice di programmi, inclusi i malware. Converte il codice binario in

linguaggio assembly, facilitando la comprensione del funzionamento interno dei programmi. Aiuta gli analisti a identificare funzioni, visualizzare il flusso di controllo e annotare il codice per comprendere le caratteristiche del malware senza eseguirlo direttamente.

PARAMETRI E VARIABILI



Utilizzando la sezione “Functions” troveremo i parametri e le variabili all’interno della funzione MAIN ()



; Attributes: bp-based frame

int __cdecl main(int argc, const char **argv, const char *envp)

_main proc near

iModule= dword ptr -11Ch

iata= byte ptr -118h

iar_8= dword ptr -8

iar_4= dword ptr -4

irgc= dword ptr 8

irgv= dword ptr 0Ch

ienvp= dword ptr 10h

push ebp

mov ebp, esp

sub esp, 11Ch

push ebx

push esi

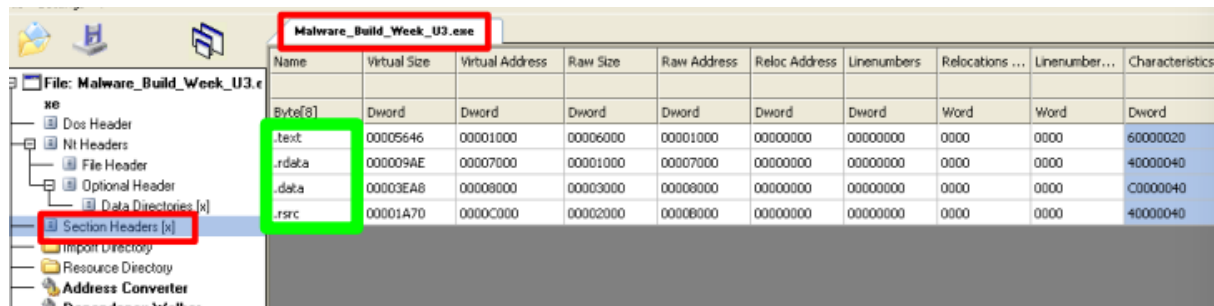
push edi

IDENTIFICAZIONE SEZIONI

Per identificazione delle sezioni ci avvalremo di CFF EXPLORER

CFF EXPLORER

CFF Explorer è un software gratuito per Windows utilizzato per esaminare e modificare file eseguibili (PE). Fornisce dettagli sulla struttura interna del programma, inclusi header, sezioni, tabelle di importazione ed esportazione. Include anche un editor esadecimale per la modifica diretta del contenuto binario. È utile per analisti di sicurezza e sviluppatori nell'analisi statica di programmi e malware.



| Name | Virtual Size | Virtual Address | Raw Size | Raw Address | Reloc Address | Linenumbers | Relocations ... | Linenumber... | Characteristics |
|--------|--------------|-----------------|----------|-------------|---------------|-------------|-----------------|---------------|-----------------|
| .text | 00005646 | 00001000 | 00006000 | 00001000 | 00000000 | 00000000 | 0000 | 0000 | 60000020 |
| .rdata | 000009AE | 00007000 | 00001000 | 00007000 | 00000000 | 00000000 | 0000 | 0000 | 40000040 |
| .data | 00003EAB | 00008000 | 00003000 | 00008000 | 00000000 | 00000000 | 0000 | 0000 | C0000040 |
| .rsrc | 00001A70 | 0000C000 | 00002000 | 00008000 | 00000000 | 00000000 | 0000 | 0000 | 40000040 |

Come si evince dall'immagine le sezioni risultano essere :

- .text
- .rdata
- .data
- .rsrc

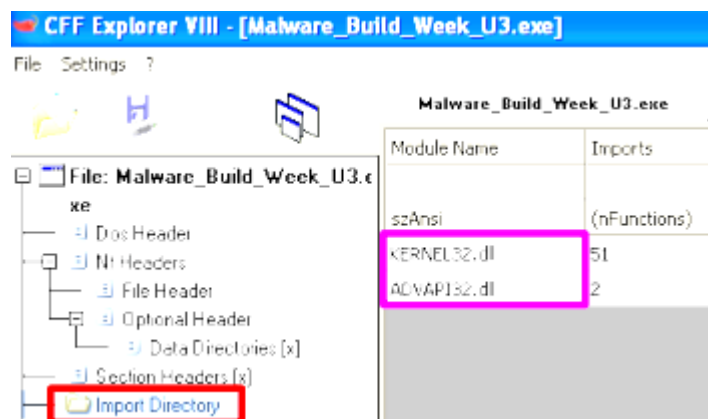
.text è quella che contiene il codice eseguibile di un programma, ed è essenziale per il funzionamento dell'eseguibile stesso.

.rdata è destinata a contenere dati di sola lettura che il programma utilizza durante l'esecuzione, e la sua natura di sola lettura implica che tali dati non vengano modificati durante l'esecuzione del programma.

.data contiene dati globali che possono essere modificati durante l'esecuzione del programma.

.rsrc è destinata a contenere le risorse non eseguibili utilizzate dal programma, offrendo un modo strutturato per archiviare elementi come immagini, stringhe e altre risorse necessarie durante l'esecuzione.

Librerie Malware



| OFTs | FTs (IAT) | Hint | Name |
|----------|-----------|------|--------------------------|
| Dword | Dword | Word | szAnsi |
| 00007632 | 00007632 | 0295 | SizeOfResource |
| 00007644 | 00007644 | 0105 | LoadResource |
| 00007654 | 00007654 | 0107 | LoadResource |
| 00007622 | 00007622 | 028B | VirtualAlloc |
| 00007674 | 00007674 | 0124 | GetModuleFileNameA |
| 0000768A | 0000768A | 0126 | GetModuleHandleA |
| 00007612 | 00007612 | 0086 | FreeResource |
| 00007664 | 00007664 | 0045 | FindResourceA |
| 00007604 | 00007604 | 001B | CloseHandle |
| 000076DE | 000076DE | 00CA | GetCommandLineA |
| 000076C0 | 000076C0 | 0174 | GetVersion |
| 000076FE | 000076FE | 007D | ExitProcess |
| 0000760C | 0000760C | 019F | HeapFree |
| 00007718 | 00007718 | 011A | GetLastError |
| 00007728 | 00007728 | 020F | WriteFile |
| 00007734 | 00007734 | 024E | TerminateProcess |
| 00007748 | 00007748 | 00F7 | GetCurrentProcess |
| 0000775C | 0000775C | 024D | UnhandledExceptionFilter |
| 00007776 | 00007776 | 008D | FreeEnvironmentStringsA |
| 00007792 | 00007792 | 0083 | FreeEnvironmentStringsW |
| 000077AC | 000077AC | 02D2 | WideCharToMultiByte |
| 000077C2 | 000077C2 | 0106 | GetEnvironmentStrings |
| 000077DA | 000077DA | 0108 | GetEnvironmentStringsW |
| 000077F4 | 000077F4 | 026D | GetHandleCount |

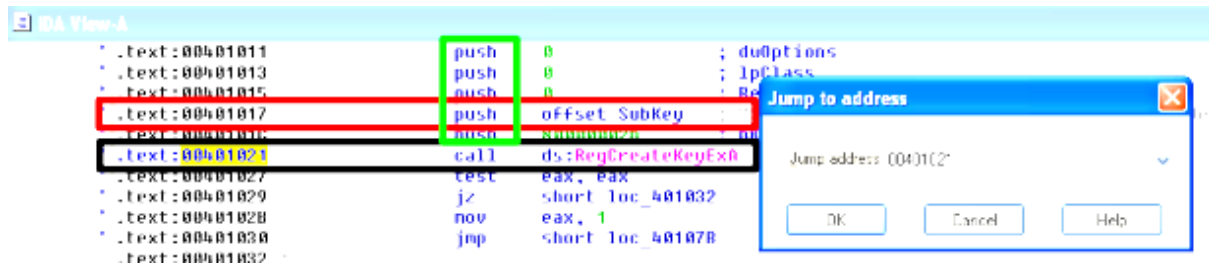
| OFTs | FTs (IAT) | Hint | Name |
|----------|-----------|----------|-------------------------|
| Dword | Dword | Word | szAnsi |
| 00007590 | 00007068 | 000077F4 | 000077F6 |
| 00007836 | 00007836 | 0109 | GetEnvironmentVariableA |
| 00007850 | 00007850 | 0175 | GetVersionExA |
| 00007860 | 00007860 | 019D | HeapDestroy |
| 0000786E | 0000786E | 0196 | HeapCreate |
| 0000787C | 0000787C | 020F | ValueFree |
| 0000788A | 0000788A | 022F | PdllLoad |
| 00007896 | 00007896 | 0199 | HeapAlloc |
| 000078A2 | 000078A2 | 01A2 | HeapReAlloc |
| 000078B0 | 000078B0 | 0C7C | SetStdHandle |
| 000078C0 | 000078C0 | 00A4 | FlushFileBuffers |
| 000078D4 | 000078D4 | 0264 | SetFilePointer |
| 000078E6 | 000078E6 | 0034 | CreateFileA |
| 000078F4 | 000078F4 | 00BF | GetFileType |
| 00007900 | 00007900 | 00B9 | GetACP |
| 0000790A | 0000790A | 0131 | GetOEMCP |
| 00007916 | 00007916 | 013E | GetProcAddress |
| 00007928 | 00007928 | 01C2 | LoadLibraryA |
| 00007938 | 00007938 | 0261 | SetEndOfFile |
| 00007948 | 00007948 | 0C16 | ReadFile |
| 00007954 | 00007954 | 01E4 | MultiByteToWideChar |
| 0000796A | 0000796A | 01BF | LCMapStringA |
| 0000797A | 0000797A | 01C0 | LCMapStringW |
| 0000798A | 0000798A | 0153 | GetStringTypeA |
| 0000799C | 0000799C | 0156 | GetStringTypeW |

| OFTs | FTs (IAT) | Hint | Name |
|----------|-----------|------|-----------------|
| Dword | Dword | Word | szAnsi |
| 000076AC | 000076AC | 0186 | RegSetValueExA |
| 000076BE | 000076BE | 015F | RegCreateKeyExA |

Funzioni Malware

Le funzioni di un malware possono includere auto-propagazione, furto di informazioni, creazione di backdoor, danni al sistema, attacchi ransomware, minacce alla sicurezza, partecipazione a botnet, attacchi DDoS, camuffamento,

auto-aggiornamento e attività di spionaggio. Il malware può variare ampiamente nel suo comportamento a seconda degli obiettivi degli attaccanti.



Analisi Dinamica

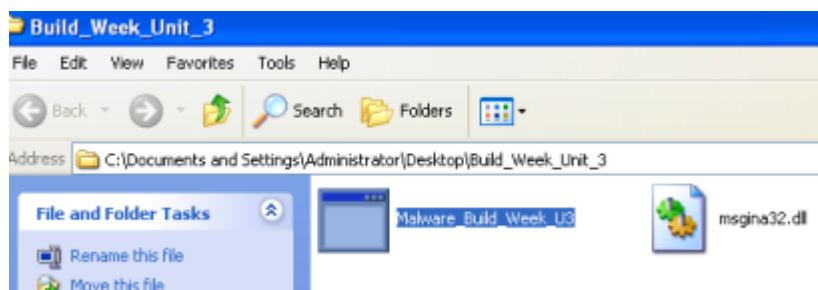
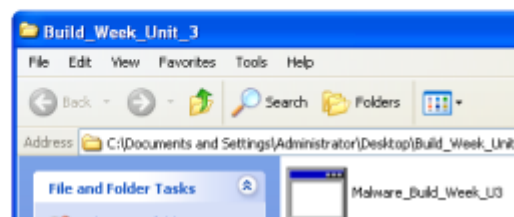
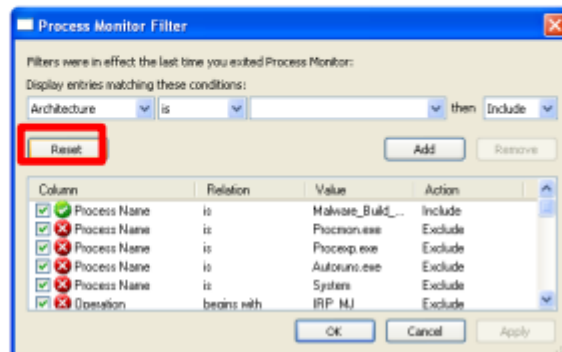
L'analisi dinamica del malware coinvolge l'esecuzione del software in un ambiente controllato per osservare il suo comportamento in tempo reale. Gli esperti monitorano le attività del malware, registrano le interazioni di rete, controllano le modifiche al sistema e identificano eventuali effetti dannosi. Questo approccio fornisce informazioni cruciali sul comportamento reale del malware e può rivelare funzionalità che non sono evidenti nell'analisi statica del codice.

Per l'esecuzione dinamica utilizzeremo Processmonitor

ProcessMonitor

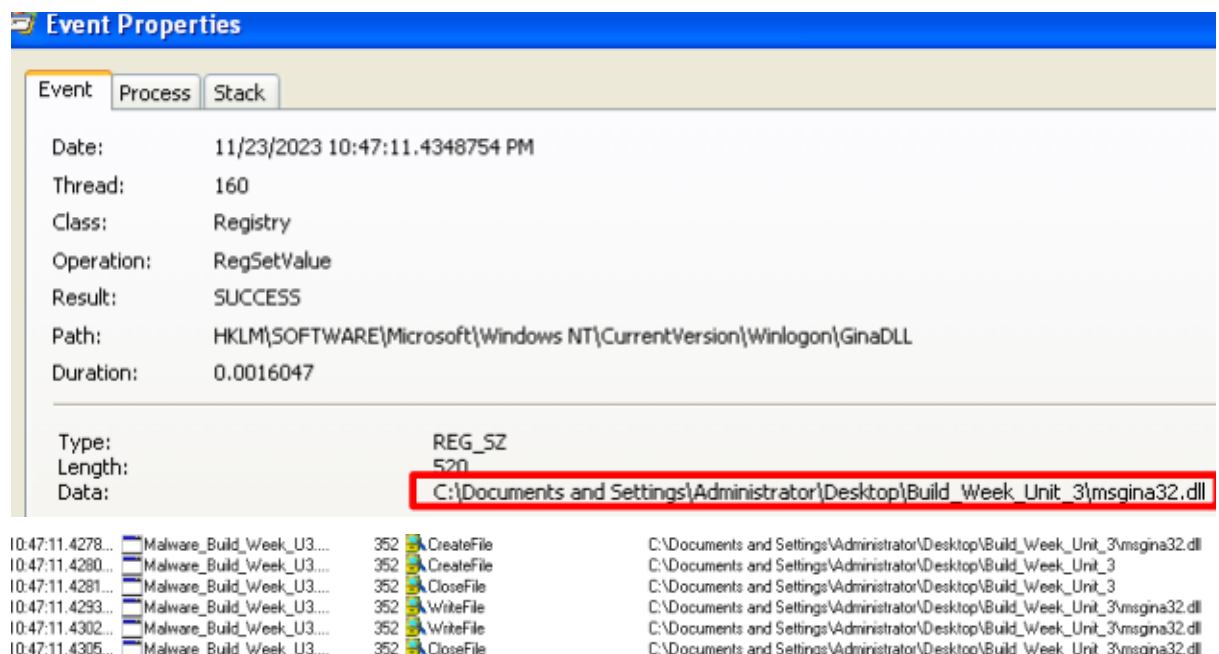
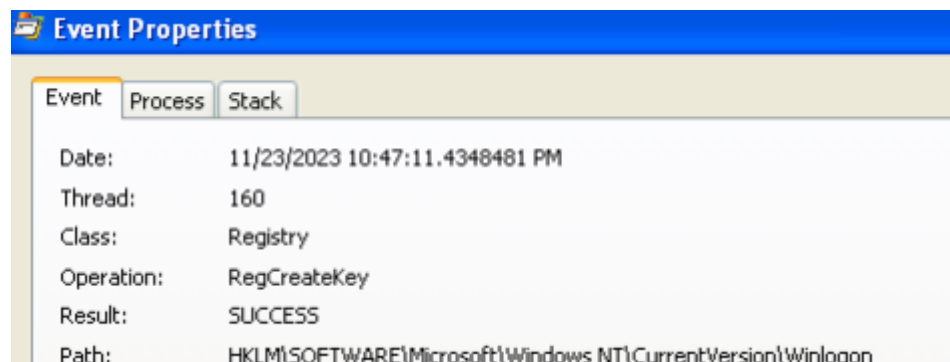
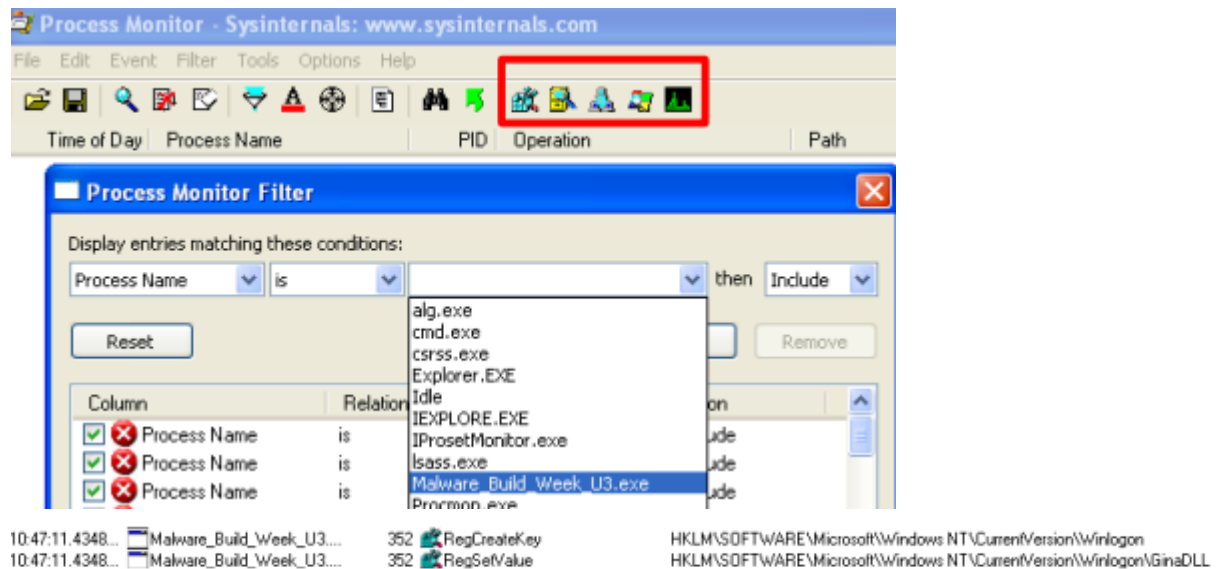
Process Monitor è uno strumento di monitoraggio del sistema per Windows.

Registra e visualizza in tempo reale le attività dei processi, inclusi accessi ai file, registri di sistema e operazioni di rete. È utilizzato per risolvere problemi, analizzare errori di applicazioni e identificare comportamenti anomali. Fornisce un'interfaccia utente intuitiva e dettagliate registrazioni delle attività del sistema.



Malware analysis

Prossimo passaggio e di controllo per andar a vedere quali processi ha eseguito



Conclusioni

Possiamo affermare con certezza che il malware sarebbe andato ad impattare la configurazione del sistema modificando la chiave di registro mediante un ulteriore malware (msgina32.dll)