TRACCIA FINALE W24D4
MALWARE ANALYSIS

TRACCIA:

IL MALWARE DA ANALIZZARE È NELLA CARTELLA **BUILD_WEEK_UNIT_3** PRESENTE SUL DESKTOP DELLA MACCHINA VIRTUALE DEDICATA.

ANALISI STATICA

CON RIFERIMENTO AL FILE ESEGUIBILE **MALWARE_BUILD_WEEK_U3**, RISPONDERE AI SEGUENTI QUESITI UTILIZZANDO I TOOL E LE TECNICHE APPRESE NELLE LEZIONI TEORICHE:

- QUANTI **PARAMETRI** SONO PASSATI ALLA **FUNZIONE MAIN()**?
- QUANTE **VARIABILI** SONO DICHIARATE ALL'INTERNO DELLA **FUNZIONE MAIN()**?
- QUALI **SEZIONI** SONO PRESENTI **ALL'INTERNO DEL FILE ESEGUIBILE**? DESCRIVETE BREVEMENTE ALMENO 2 DI QUELLE IDENTIFICATE
- **QUALI LIBRERIE** IMPORTA IL MALWARE? PER OGNUNA DELLE LIBRERIE IMPORTATE, FATE DELLE IPOTESI SULLA BASE DELLA SOLA ANALISI STATICA DELLE FUNZIONALITÀ CHE IL MALWARE POTREBBE IMPLEMENTARE. UTILIZZATE LE FUNZIONI CHE SONO RICHIAMATE ALL'INTERNO DELLE LIBRERIE PER SUPPORTARE LE VOSTRE IPOTESI.

L'ESECUZIONE DELL'ESERCIZIO RICHIEDE LA COMBINAZIONE DI VARIE ANALISI VISTE DURANTE IL MODULO 6 DEL CORSO.
PER QUESTA PRIMA PARTE SERVIRÀ L'APPLICAZIONE DELL'ANALISI STATICA BASICA ED AVANZATA.
L'ANALISI STATICA SI RIFERISCE ALL'ISPEZIONE DEL CODICE SORGENTE O DEL CODICE BINARIO DI UN PROGRAMMA (IN QUESTO CASO, UN MALWARE) PER IDENTIFICARNE LA FUNZIONALITÀ, LE CARATTERISTICHE E LE POTENZIALI MINACCE SENZA ESEGUIRLO. QUESTO APPROCCIO SI CONTRAPPONE ALL'ANALISI DINAMICA, DOVE IL CODICE VIENE ESEGUITO IN UN AMBIENTE CONTROLLATO (SANDBOX) PER OSSERVARE IL SUO COMPORTAMENTO.

L'ANALISI STATICA BASICA CONSISTE NELL'ESAMINARE UN ESEGUIBILE SENZA VEDERE LE ISTRUZIONI CHE LO COMPONGONO E LA SUA FUNZIONE È CONFERMARE SE UN DATO FILE È MALEVOLO E FORNIRE INFORMAZIONI GENERICHE CIRCA LE SUE FUNZIONALITÀ. L'ANALISI STATICA AVANZATA PRESUPPONE LA CONOSCENZA DEI FONDAMENTI DI «REVERSE-ENGINEERING» AL FINE DI IDENTIFICARE IL COMPORTAMENTO DI UN MALWARE A PARTIRE DALL'ANALISI DELLE ISTRUZIONI CHE LO COMPONGONO. QUESTO PASSAGGIO È ESSENZIALE PER CAPIRE ESATTAMENTE COSA FA IL MALWARE A LIVELLO DI ISTRUZIONI DELLA CPU. SI POSSONO INOLTRE ESTRARRE STRINGHE DI TESTO, URL, CHIAVI DI CIFRATURA, E ALTRE RISORSE DAL CODICE DEL MALWARE, CHE POSSONO INDICARE IL SUO COMPORTAMENTO O INTENTO E SE NE PUÒ ESAMINARE IL CODICE RELATIVO ALLA RETE PER COMPRENDERE COME IL MALWARE COMUNICA.

PRIMA DI PROCEDERE ALL'ANALISI MI ASSICURO CHE SIA DI FATTO UN MALWARE, ESTRAENDONE L'HASH CON MD5DEEP E CONTROLLANDO SU VIRUSTOTAL LA SUA REPUTAZIONE CHE SI BASA SU VARI RISCONTRI DI SOFTWARE ANTIVIRUS.

IN QUESTA PRIMA ANALISI POSSO VEDERE CHE IL VIRUS è NOTO, SI TRATTA DI UN MALWARE DI TIPO TROJAN COMPILATO IN DATA 11-06-2011 IN C++, ANALIZZATO L'ULTIMA VOLTA IN DATA 17 APRILE 2024. È UN MALWARE PROGETTATO PER COLPIRE LA MACCHINA INTEL 386 E PROCESSORI SUCESSIVI/COMPATIBILI.
HA 5255 ENTRY POINTS, 4 SEZIONI ED IMPORTA 2 LIBRERIE:
KERNEL32.DLL - UNA DELLE LIBRERIE FONDAMENTALI DI WINDOWS, CONTIENE NUMEROSE FUNZIONI CHE GESTISCONO LA MEMORIA, I PROCESSI E I THREAD. I MALWARE LA UTILIZZANO PER MANIPOLARE I PROCESSI E PER ACCEDERE A DIVERSE API DI SISTEMA ED OTTENERE PERSISTENZA.
ADVAPI32.DLL - FORNISCE FUNZIONI RELATIVE ALLA SICUREZZA E ALLA GESTIONE DI ACCOUNT, CHE I MALWARE POSSONO SFRUTTARE PER MODIFICARE PERMESSI, ACCEDERE A TOKEN DI SICUREZZA E ALTERARE IL REGISTRO DI SISTEMA, AD ESEMPIO PER ESSERE AVVIATI ALL'AVVIO DEL SISTEMA OPERATIVO.

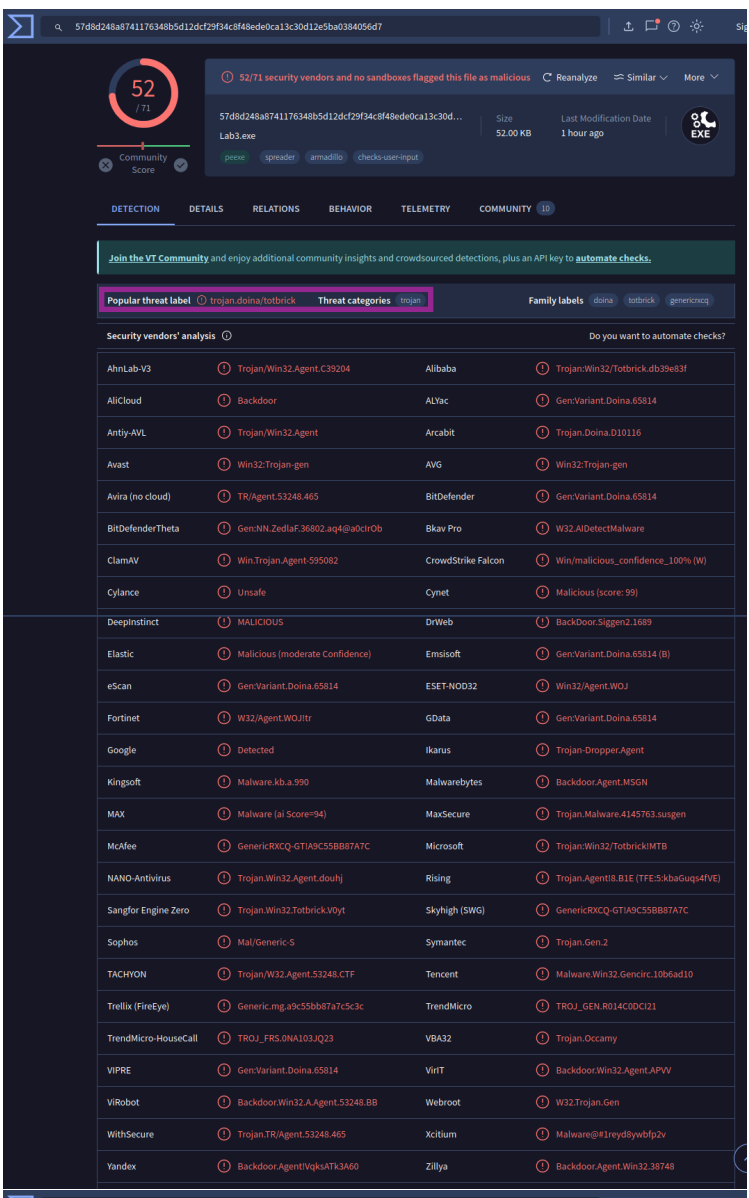
ANDRÒ A CONFERMARE TUTTI QUESTI DATI TRAMITE TOOL DEDICATI COME CFFEXPLORER/EXEINFOPE E IDA PRO.

#HASH CON MD5DEEP64

```
C:\Users\user\Desktop\Software Malware analysis\md5deep-4.3>cd md5deep-4.3

C:\Users\user\Desktop\Software Malware analysis\md5deep-4.3\md5deep-4.3>md5deep64.exe -Malware_Build_Week_U3.exe
a9c55bb87a7c5c3c923c4fa12940e719  C:\Users\user\Desktop\Software Malware analysis\md5deep-4.3\md5deep-4.3\Malware_Build_Week_U3.exe
```

#VIRUSTOTAL



CFF EXPLORER:

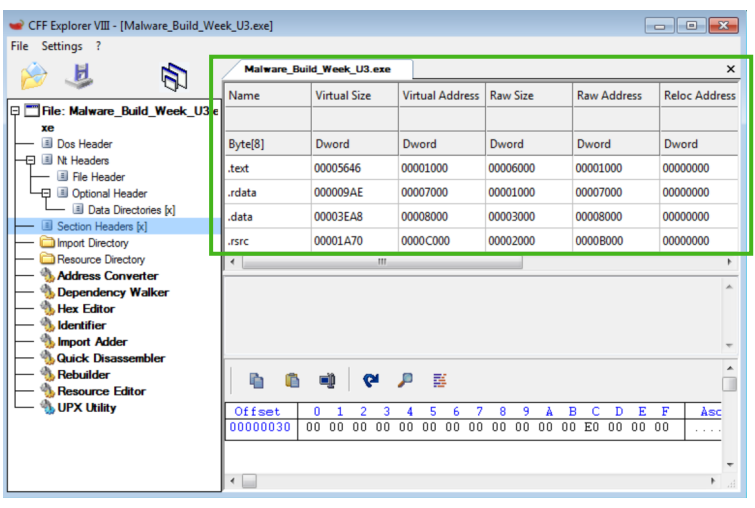
PER CONTROLLARE LE FUNZIONI IMPORTATE ED ESPORTATE DA UN MALWARE, SI PUÒ UTILIZZARE CFF EXPLORER, UN TOOL DA INSTALLARE SULLE MACCHINE VIRTUALI DEDICATE ALL'ANALISI DEI MALWARE. SE MI SPOSTO SU «IMPORT DIRECTORY» POSSO CONTROLLARE LE LIBRERIE E LE FUNZIONI IMPORTATE, MENTRE SU «SECTION HEADERS» POSSO VEDERE LE SEZIONI PRESENTI ALL'INTERNO DEL FILE ESEGUIBILE:

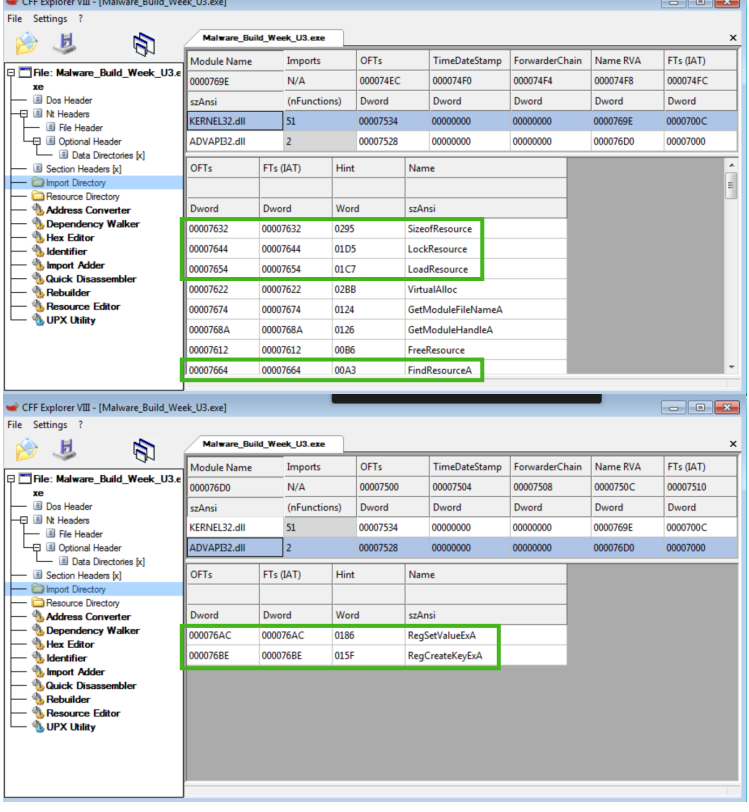
.TEXT - CONTIENE LE RIGHE DI CODICE, ISTRUZIONI, CHE LA CPU ESEGUIRÀ ALL'AVVIO DEL MALWARE.

.RDATA - SEZIONE CHE INCLUDE LE INFORMAZIONI CIRCA LE LIBRERIE E LE FUNZIONI IMPORTATE ED ESPORTATE DALL'ESEGUIBILE, DATI CHE IL PROGRAMMA LEGGE MENTRE È IN FUNZIONE.

.DATA - CONTIENE I DATI / LE VARIABILI GLOBALI DEL PROGRAMMA ESEGUIBILE, CHE DEVONO ESSERE DISPONIBILI DA QUALSIASI PARTE DEL PROGRAMMA E POSSONO ESSERE MODIFICATI.

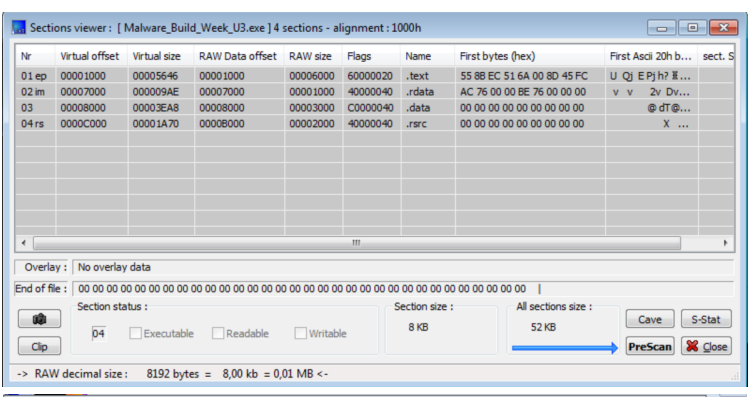
.RSRC - INCLUDE LE RISORSE UTILIZZATE DALL'ESEGUIBILE COME AD ESEMPIO ICONE, IMMAGINI, MENU E STRINGHE CHE NON SONO PARTE DELL'ESEGUIBILE STESSO.

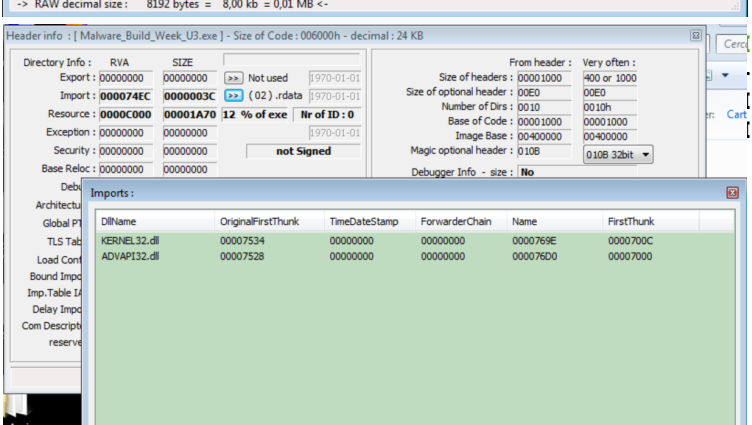




PER QUEL CHE RIGUARDA LE DUE LIBRERIE IMPORTATE, CHE SONO KERNEL32.DLL (1) E ADVAPI32.DLL (2), POSSO IPOTIZZARE CHE IL MALWARE CERCHI DI OTTENRE PERSISTENZA E MODIFICARE LE CHIAVI DI REGISTRO (REGSETVALUEEXA/REGCREATEKEYEXA) PER POTER ESSERE ESEGUITO IN AUTONOMIA. INOLTRE LA PRESENZA DI FUNZIONI COME SIZEOFRESOURCE/LOCKRESOURCE/LOADRESOURCE/FINDRESOURCEA FA PRESUPPORRE CHE SIA UN DROPPER, CIOÈ UN MALWARE CHE CONTIENE AL SUO INTERNO UN ALTRO MALWARE, ED UTILIZZA QUESTE APIS CHE PERMETTONO DI LOCALIZZARE ALL'INTERNO DELLA SEZIONE «RISORSE» IL MALWARE DA ESTRARRE, E SUCCESSIVAMENTE DA CARICARE IN MEMORIA PER L'ESECUZIONE.

EXEINFOPE: STESSI DATI CONFERMATI





IDAPRO:

ORA PER INFORMAZIONI SU VARIABILI LOCALI E PARAMETRI DELLA FUNZIONE MAIN (), USERò IDA PRO (INTERACTIVE DISASSEMBLER PROFESSIONAL) CHE È UNO STRUMENTO AVANZATO DI REVERSE ENGINEERING SOFTWARE CHE OFFRE CAPACITÀ DI DISASSEMBLAGGIO, DEBUGGING E ANALISI STATICA.

VARIABILI LOCALI: LE VARIABILI LOCALI IN UNA FUNZIONE ASSEMBLY SONO TIPICAMENTE ALLOCATE NELLO STACK. QUESTO È SPESSO FATTO ATTRAVERSO ISTRUZIONI DI TIPO PUSH ALL'INIZIO DI UNA FUNZIONE O CON UN'ISTRUZIONE SUB CHE AUMENTA IL PUNTATORE DELLO STACK (SP) PER CREARE SPAZIO.

PARAMETRI: SOLITAMENTE I PARAMETRI SONO PASSATI AI REGISTRI O ATTRAVERSO L'USO DELLO STACK PRIMA DELLA CHIAMATA DELLA FUNZIONE. I COMMENTI NEL CODICE POSSONO FORNIRE INDICAZIONI SU DOVE E QUANTI PARAMETRI SONO PASSATI.

NELL'ASSEMBLY, LE ETICHETTE CHE INIZIANO CON VAR_ TENDONO A INDICARE VARIABILI LOCALI, MENTRE QUELLE CHE INIZIANO CON ARG_ INDICANO ARGOMENTI O PARAMETRI PASSATI ALLA FUNZIONE.

VALORI OFFSET: GLI OFFSET (COME VAR_54H) SONO UTILIZZATI PER ACCEDERE A DATI SPECIFICI SULLO STACK. GLI OFFSET NEGATIVI RISPETTO ALL'INDIRIZZO BASE DEL FRAME DELLA FUNZIONE (AD ESEMPIO EBP SU X86) DI SOLITO INDICANO VARIABILI LOCALI, MENTRE GLI OFFSET POSITIVI INDICANO PARAMETRI.

IN QUESTO CASO LE VARIABILI LOCALI IN EVIDENZA CHE POSSIAMO RILEVARE AD UN'OCCHIATA SONO 5: hModule / Data / var_117 / var_8 / var_4.
MENTRE I PARAMETRI SONO 3 - argc / argv / envp.