W24D4 - Progetto

Malware Analysis

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.

Con riferimento al Malware in analisi, spiegare:

- Lo scopo della funzione chiamata alla locazione di memoria 00401021
- Come vengono passati i parametri alla funzione alla locazione 00401021;
- Che oggetto rappresenta il parametro alla locazione 00401017
- Il significato delle istruzioni comprese tra gli indirizzi 00401027 e 00401029.
- Con riferimento all'ultimo quesito, tradurre il codice Assembly nel corrispondente costrutto C.
- Valutate ora la chiamata alla locazione 00401047, qual è il valore del parametro «ValueName»?

• Per verificare quali parametri e variabili sono dichiarati utilizziamo IDApro. I parametri passanti per la funzione Main() con offset positivo sono : argc , argv e envp .

• Mentre le variabili dichiarate nella medesima con offset negativo sono : hModule , Data , var_117 , var_8 e var_4 .

```
; int __cdecl main(int argc, const char **argv, const char **envp)
_main proc near

hModule= dword ptr -11Ch
Data= byte ptr -118h
var_117= byte ptr -117h
var_8= dword ptr -8
var_4= dword ptr -4
argc= dword ptr  8
argv= dword ptr  0Ch
envp= dword ptr  10h
```

Possiamo individuare le sezioni tramite CFF Explorer e sono :

- .text : che contiene il codice eseguibile del programma .
- .rdata : che contiene informazioni sulla lettura dei dati .
- .data : che contiene dati inizializzati e non inizializzati .
- .rsrc : che contiene tools di personalizzazione come icone e menu

Malware_Build_Week_U3.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	00005646	00001000	00006000	00001000	00000000	00000000	0000	0000	60000020
.rdata	000009AE	00007000	00001000	00007000	00000000	00000000	0000	0000	40000040
.data	00003EA8	00008000	00003000	00008000	00000000	00000000	0000	0000	C0000040
.rsrc	00001A70	0000C000	00002000	0000B000	00000000	00000000	0000	0000	40000040

Il malware importa le librerie di ADVAPI32.dll e KERNEL32.dll .

- ADVAPI32.dll è una libreria utilizzata per gestione del registro di sistema e sicurezza . Il malware potrebbe utilizzarla per modificare impostazioni del registro e sulla sicurezza del sistema .
- La funzione importata è quella di RegCreateKeySetA che gli permette di creare una chiave di registro o aprire una chiave esistente all'interno del nostro registro di sistema .
- KERNEL32.dll è una libreria che permette di accedere a funzioni di memoria e quindi manipolare memoria e/o eseguire diverse task .

```
.text:00401017                push    offset SubKey   ; "SOFTWARE\\Microsoft\\Windows NT\\CurrentVe"...
.text:0040101C                push    80000002h       ; hKey
.text:00401021                call    ds:RegCreateKeyExA
```

- La locazione di memoria 00401021 ha come funzione RegCreateKeyExa che il compito di creare una chiave di registro i parametri gli vengono passati attraverso la funzione push .
- Alla funzione 00401017 c'è l'offset Subkey che identifica la chiave di registro stessa .

- Tra le funzioni 00401027 e 00401029 troviamo istruzioni per un salto condizionale in cui se la funzione eax è uguale a 0 avviene un salto alla locazione 00401032 con uno jump zero .

- Istruzioni : test eax, eax
  jz short loc_401032

- La stessa funzione espressa in linguaggio C :

```c
#include <stdio.h>

void function(int ax) {
    if (eax == 0) {
        goto zero_case;
    }

    // Next action
    printf("Continuing with next action...\n");
    return;

zero_case:
    // Jump to address 401032
    printf("Jumping to address 401032...\n");
    goto *((void*)0x401032);
}

int main() {
    function(0);  // Test with eax = 0
    function(1);  // Test with eax!= 0
    return 0;
}
```

- Avviando il malware notiamo che inseguito viene creato il file Gina.dll questo ha lo scopo di modificare il registro di sistema sostituendo una chiave nel processo di autenticazione di Windows che potrebbe consentire al malintenzionato accesso e raccolta di informazioni e credenziali sul nostro sistema .