Report sull'Analisi del Malware Utilizzando IDA Pro

Data 14/02/24

Introduzione

L'obiettivo dell'esercizio odierno era di acquisire esperienza nell'utilizzo di IDA, uno strumento fondamentale per l'analisi statica dei malware. In particolare, è stato richiesto di analizzare il malware denominato "Malware_U3_W3_L2", presente all'interno della cartella "Esercizio_Pratico_U3_W3_L2" sul desktop della macchina virtuale dedicata all'analisi dei malware. Utilizzando IDA Pro, sono stati esaminati diversi aspetti del malware, come la localizzazione della funzione DLLMain, l'individuazione di funzioni specifiche e la determinazione di variabili e parametri all'interno del codice.

Analisi dei Quesiti

Individuazione dell'Indirizzo della Funzione DLLMain

La funzione DLLMain è una funzione di callback eseguita quando un modulo DLL viene caricato o scaricato, o quando si verifica un evento specifico all'interno della DLL. Utilizzando IDA Pro, è stato individuato l'indirizzo della funzione DLLMain all'interno del malware "Malware_U3_W3_L2".

```
Function name
                                                   .text:1000D02E ; BOOL
🛐 sub_1000C8EA
                                                   .text:1000D02E
.text:1000D02E
.text:1000D02E
                                                                                                                        ; CODE XREF: D11EntryPoint+4Blp
; DATA XREF: sub_100110FF+2Dlo
📆 sub_1000CA56
sub_1000CC06
                                                                                          = dword ptr
= dword ptr
= dword ptr
                                                   .text:1000D02E hinstDLL
🛐 ServiceMain
                                                   .text:1000D02E fdwReason
.text:1000D02E lpvReserved
DllMain(x,x,x)
                                                                                                          OCh
                                                    .text:1000D02E
.text:1000D02E
Fisub 1000D10D
                                                                                                    eax, [esp+fdwReason]
📆 sub_1000D1D3
                                                   .text:1000D032
                                                                                          dec
sub_1000D26B
                                                   .text:1000D033
                                                                                                    loc_1000D107
eax, [esp+hinstDLL]
nsub_1000D2F9
🛐 sub_1000D3D0
                                                   .text:1000D03D
                                                                                          push
                                                                                                    ebx
                                                                                                    ds:hModule, eax
eax, off_10019044
🕅 sub 1000D5B0
                                                   .text:1000D03E
InstallRT
```

L'indirizzo della funzione DLLMain corrisponde a :

- Text: 1000D02E

Abbiamo recuperato l'indirizzo di memoria cercando tra le varie funzioni presenti nella finestra a sinistra del programma. La sezione si chiasma appunto "Functions name".

Individuazione dell'Indirizzo dell'Import della Funzione "gethostbyname"

La funzione gethostbyname è una funzione della libreria di sistema del linguaggio di programmazione C utilizzata per ottenere informazioni sulle entità di rete, come gli host, utilizzando il loro nome. Questa funzione traduce il nome di un host in una struttura di tipo hostent, che contiene informazioni come l'indirizzo IP associato all'host.

Utilizzando la scheda "imports" di IDA Pro, è stata individuata la funzione "gethostbyname" all'interno delle importazioni del malware. È stata identificata l'indirizzo "idata:100163CC" dell'import relativo a questa funzione.

```
.idata:100163C4
                                                           ; CODE XREF: sub_10001656+3D21p
                                  extrn select:dword
 .idata:100163C8 ; unsigned __int32
                                  ..data:100163C8 ; sub_10001074+1BF†p ...
Lidata:100163CC ; struct hostent * stdcall qethostbyname(const char *name)
                                  extrn gethostbyname:dword
                                                           : CODE XREF: sub 10001074:loc_100011AFTp
                                                             sub_10001074+1D31p ..
 .idata:100163D0 ; char *__stdcall inet_ntoa(struct in_addr in)
                                  extrn inet_ntoa:dword ; CODE XREF: sub_10001074:loc_100013111p
                                                            ; sub_10001365:1oc_10001602<sup>†</sup>p
 .idata:100163D4 ; int __stdcall recv(SOCKET s, char *buf,
                                                             int len, int flags)
CODE XREF: sub_10001656+2D51p
                                  extrn recv:dword
                                                             sub_10001656+3F21p ...
*buf, int len, int flags)
 .idata:100163D4
  idata:100163D8 ; int __stdcall send(SOCKET s, const char
                                                           ; CODE XREF: sub_10001656+2AB1p
```

Abbiamo notato che questo indirizzo di memoria corrisponde alla definizione della funzione ma non sono presenti i vari parametri di cui essa ha bisogno. Un'altra particolarità è la chiamata a questa funzione all'interno della funzione DLLMain dove oltre alla chiamata della funzione "gethostbyname" sono presenti una serie di parametri che vengono impostati prima della chiamata a questa funzione.

Questo è un punto chiave dell'analisi, in quanto potrebbe indicare che i parametri necessari per la funzione gethostbyname vengono preparati o inizializzati prima della sua chiamata all'interno della funzione DLLMain. Questa pratica è comune nel codice di basso livello, dove le funzioni possono dipendere da variabili o dati esterni che devono essere preparati prima dell'esecuzione della funzione stessa.

Determinazione delle Variabili Locali alla Locazione di Memoria 0x10001656

Analizzando la funzione o le funzioni associate alla locazione di memoria specificata, sono state determinate le variabili locali presenti in quel contesto.

Conteggio dei Parametri della Funzione:

Utilizzando le informazioni fornite da IDA Pro, è stato contato il numero di parametri della funzione specificata. In base a quanto abbiamo appreso a lezione possiamo distinguere i parametri dalle variabili in base al segno del valore corrispondente. In questo caso i parametri corrispondono ai valori che presentano segno positivo. In base alla lista trovata a quell'indirizzo di memoria abbiamo trovato un solo parametro "arg_0".

```
.text:10001656 var_675
                               = byte ptr -675h
.text:10001656 var_674
                               = dword ptr -674h
.text:10001656 hLibModule
                               = dword ptr -670h
.text:10001656 timeout
                               = timeval ptr -66Ch
.text:10001656 name
                               = sockaddr ptr -664h
.text:10001656 var_654
                               = word ptr -654h
.text:10001656 Dst
                               = dword ptr -650h
.text:10001656 Parameter
                               = byte ptr -644h
.text:10001656 var_640
                               = byte ptr -640h
.text:10001656 CommandLine
                               = byte ptr -63Fh
.text:10001656 Source
                               = byte ptr -63Dh
.text:10001656 Data
                               = byte ptr -638h
.text:10001656 var_637
                               = byte ptr -637h
                               = dword ptr -544h
.text:10001656 var_544
.text:10001656 var_50C
                               = dword ptr -50Ch
.text:10001656 var_500
                               = dword ptr -500h
.text:10001656 Buf2
                               = byte ptr -4FCh
.text:10001656 readfds
                               = fd_set ptr -4BCh
.text:10001656 phkResult
                               = byte ptr -3B8h
.text:10001656 var_3B0
                               = dword ptr -3B0h
.text:10001656 var_1A4
                               = dword ptr -1A4h
.text:10001656 var_194
                               = dword ptr -194h
.text:10001656 WSAData
                               = WSAData ptr -190h
.text:10001656 arg_0
                               = dword ptr
```

Conclusioni

L'analisi condotta utilizzando IDA Pro ha fornito una panoramica dettagliata del malware "Malware_U3_W3_L2". Sono stati individuati e risolti diversi quesiti relativi al suo funzionamento interno, inclusi l'indirizzo della funzione DLLMain, l'import della funzione "gethostbyname", il numero di variabili locali e il conteggio dei parametri delle funzioni coinvolte. Questo esercizio ha contribuito ad aumentare la comprensione delle tecniche di analisi statica dei malware e all'uso efficace degli strumenti come IDA Pro.