

ANALISI STATICÀ AVANZATA CON

IDA

Mattia Pastorelli

COMANDA:

Lo scopo dell'esercizio di oggi è di acquisire esperienza con IDA, un tool fondamentale per l'analisi statica. A tal proposito, con riferimento al malware chiamato «Malware_U3_W3_L2» presente all'interno della cartella «Esercizio_Pratico_U3_W3_L2» sul Desktop della macchina virtuale dedicata all'analisi dei malware, rispondere ai seguenti quesiti, utilizzando IDA Pro.

1. Individuare l'indirizzo della funzione DLLMain(così com'è, in esadecimale)
2. Dalla scheda «imports» individuare la funzione «gethostbyname». Qual è l'indirizzo dell'import? Cosa fa la funzione?
3. Quante sono le variabili locali della funzione alla locazione di memoria 0x10001656?
4. Quanti sono, invece, i parametri della funzione sopra? 5. Inserire altre considerazioni macro livello sul malware (comportamento)

INDIVIDUARE L'INDIRIZZO DELLA FUNZIONE DLLMAIN (COSÌ COM'È, IN ESADECIMALE)

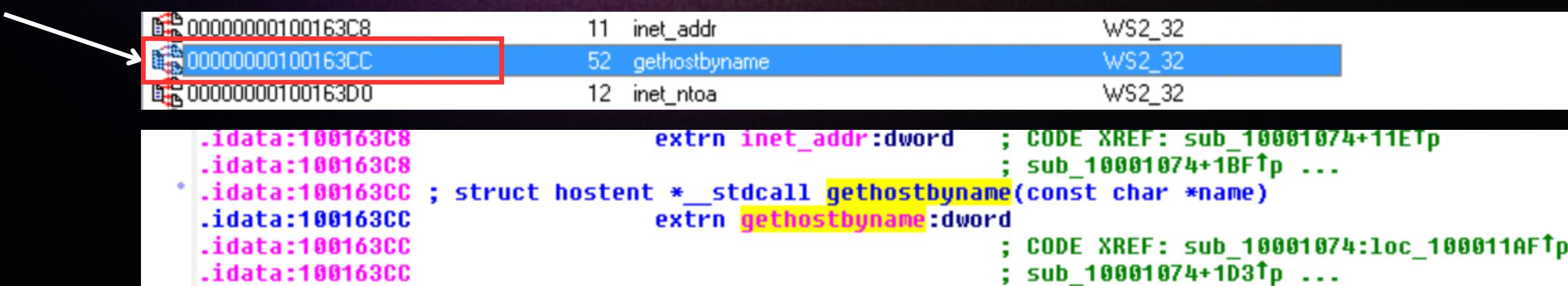
```
; BOOL __stdcall DllMain(HINSTANCE hinstDLL, DWORD FdwReason, LPVOID lpvReserved)
_DllMain@12 proc near

hinstDLL= dword ptr  4
FdwReason= dword ptr  8
lpvReserved= dword ptr  0Ch
```

1000D01E	F8 03 74 05 83 F8 01 75 E9 5F 5E 5B C9 C2 08 00	°.t
1000D02E	8B 44 24 08 48 0F 85 CE 00 00 00 8B 44 24 04 53	iD\$
1000D03E	A3 00 30 09 10 A1 44 90 01 10 56 83 C0 0D 57 50	ú.º
1000D04E	E8 F9 7E 00 00 8B 1D 08 62 01 10 8B 35 C0 62 01	þ~
1000D05E	48 00 FF F0 05 00 71 00 41 11 00 01 10 0A 07 00	?

DALLA SCHEDA «IMPORTS» INDIVIDUARE LA FUNZIONE «GETHOSTBYNAME». QUAL È L'INDIRIZZO DELL'IMPORT? COSA FA LA FUNZIONE?

Indirizzo



Indirizzo	Nome simbolico	Libreria
00000000100163C8	11 inet_addr	WS2_32
00000000100163CC	52 gethostbyname	WS2_32
00000000100163D0	12 inet_ntoa	WS2_32


```
.idata:100163C8          extrn inet_addr:dword ; CODE XREF: sub_10001074+11ETp
.idata:100163C8          ; sub_10001074+1BF↑p ...
* .idata:100163CC ; struct hostent * __stdcall gethostbyname(const char *name)
.idata:100163CC          extrn gethostbyname:dword
.idata:100163CC          ; CODE XREF: sub_10001074:loc_100011AF↑p
.idata:100163CC          ; sub_10001074+1D3↑p ...
```

Attraverso questa funzione, il malware cerca di ottenere l'indirizzo IP della possibile vittima, al fine di ottenere una connessione di rete.

QUANTE SONO LE VARIABILI LOCALI DELLA FUNZIONE ALLA LOCAZIONE DI MEMORIA oX10001656?

Totale di 23 funzioni in questa locazione

10001646	00	59	50	FF	15	1C	62	01	10	33	ED	E9	4F	FD	FF	FF	.YP ..B.
10001656	81	EC	78	06	00	0E	53	55	56	57	E8	9B	F9	FF	FF	85	üýx...SU
10001666	C0	75	53	33	DB	89	5C	24	14	89	5C	24	18	E8	1D	20	+uS3!ë\\$
10001676	00	00	00	00	EE	00	10	50	14	00	00	00	10	00	00	00	í ö

var_675= byte ptr -675h
var_674= dword ptr -674h
hLibModule= dword ptr -670h
timeout= timeval ptr -66Ch
name= sockaddr ptr -664h
var_654= word ptr -654h
Dst= duord ptr -650h
Parameter= byte ptr -644h
var_640= byte ptr -640h
CommandLine= byte ptr -63Fh
Source= byte ptr -63Dh
Data= byte ptr -638h
var_637= byte ptr -637h
var_544= dword ptr -544h
var_50C= dword ptr -50Ch
var_500= dword ptr -500h
Buf2= byte ptr -4FCh
readFds= fd_set ptr -48Ch
phkResult= byte ptr -388h
var_3B0= dword ptr -3B0h
var_1A4= dword ptr -1A4h
var_194= dword ptr -194h
WSAData= WSADATA ptr -190h
arg_0= dword ptr 4

QUANTI SONO, INVECE, I PARAMETRI DELLA FUNZIONE SOPRA?

```
10001646  00 59 50 FF 15 1C 62 01  10 33 ED E9 4F FD FF FF  .YP ..D..3Y00<
10001656  81 EC 78 06 00 00 53 55  56 57 E8 9B F9 FF FF 85  üýx...SUUWPø" à
10001666  C0 75 53 33 DB 89 5C 24  14 89 5C 24 18 E8 1D 20  +uS3!ë\$.ë\$.p.
10001676  00 00 00 01 FF 00 10 F0  14 00 00 00 F0 00 00 00  é ï 10 00 00 00
```

I Parametri sono 4

```
var_675= byte ptr -675h
var_674= dword ptr -674h
hLibModule= dword ptr -670h
timeout= timeval ptr -66Ch
name= sockaddr ptr -664h
var_654= word ptr -654h
Dst= duord ptr -650h
Parameter= byte ptr -644h
var_640= byte ptr -640h
CommandLine= byte ptr -63Fh
Source= byte ptr -630h
Data= byte ptr -638h
var_637= byte ptr -637h
var_544= dword ptr -544h
var_58C= dword ptr -580h
var_500= dword ptr -500h
Buf2= byte ptr -4FCh
readfds= fd_set ptr -48Ch
phkResult= byte ptr -3B8h
var_3B0= dword ptr -3B0h
var_1A4= dword ptr -1A4h
var_194= dword ptr -194h
WSAData= WSAData ptr -190h
arg_0= dword ptr 4
```

INSERIRE ALTRE CONSIDERAZIONI MACRO LIVELLO SUL MALWARE (COMPORTAMENTO)

Il programma ha lo scopo di inserire una backdoor nel computer della vittima per poter accedere ai dati in qualsiasi momento.

Utilizzando le funzioni "GetHostByName" e "Recv", il malware cerca di ricevere i dati attraverso la connessione di rete.

Attraverso la funzione BackdoorServer, viene creato un server in ascolto su una porta specifica, pronto ad accettare connessioni in arrivo.

Dopo aver infettato il computer, il malware entra in modalità "Sleep" per evitare di essere rilevato dagli antivirus/utente.

```
loc_10006F87:          ; Size
push    8
lea     eax, [ebp+Dst]
push    0
push    eax           ; Val
push    eax           ; Dst
call    memset
add    esp, 0Ch
lea     eax, [ebp+Dst]
push    0             ; Flags
push    8             ; len
push    eax           ; buf
push    esi           ; s
call    ds:recv
cmp    eax, 0FFFFFFFh
jz     short loc_10007017

push    eax           ; lpBuffer
push    edi           ; nBufferLength
call    ds:GetCurrentDirectoryA
mov    esi, ds:sprintf
lea     eax, [ebp+buf]
push    offset aBackdoorServer ; "\r\n\r\n*****"
push    eax           ; Dest
call    esi ; sprintf
mov    ebx, [ebp+s]
lea     eax, [ebp+buf]
push    eax           ; buf
push    ebx           ; s
call    sub_100038BB
add    esp, 10h
lea     eax, [ebp+PathName]
push    eax           ; lpPathName
call    ds:SetCurrentDirectoryA
test   eax, eax
jz     loc_100046E1
```

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label ! trojan.idicaf/r06cc0df321 Threat categories trojan Family labels idicaf r06cc0df321

Security vendors' analysis !

AhnLab-V3	! Backdoor/Win32.Agent.R9408	Alibaba	! Backdoor:Win32/Idicaf.9f3a5556
ALYac	! Backdoor.XIW	Antiy-AVL	! Trojan[Backdoor]/Win32.Agent
Arcabit	! Backdoor.XIW	Avast	! Win32:Agent-OLH [Trj]
AVG	! Win32:Agent-OLH [Trj]	Avira (no cloud)	! BDS/Agen.twe.134160

Scansione con virustotal