

## Esercizio 2 Settimana 11

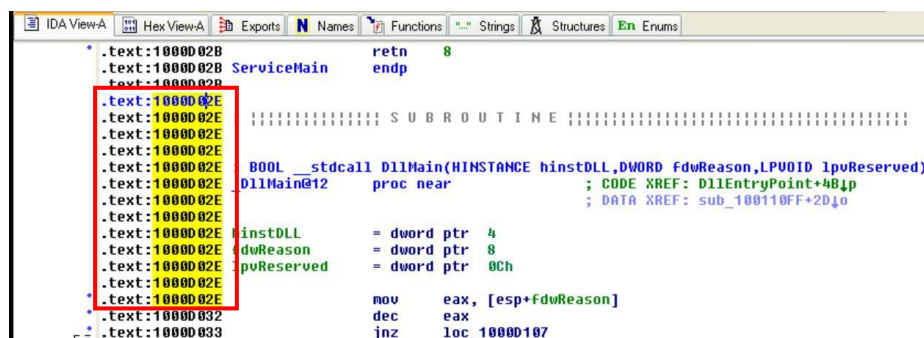
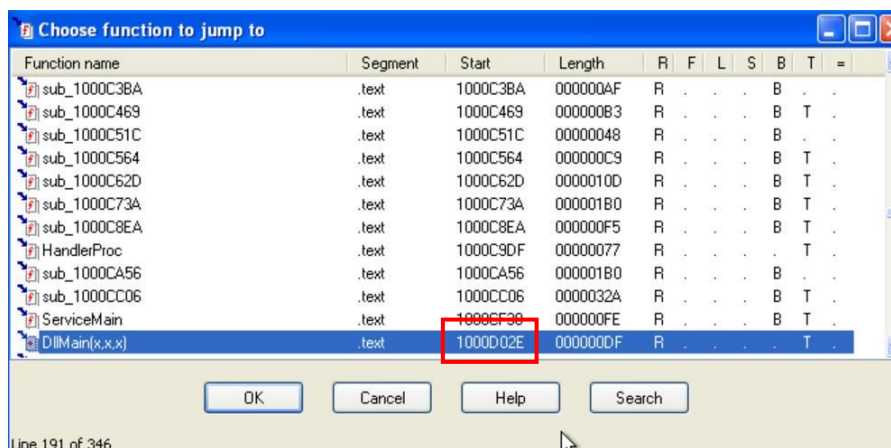
## Analisi statica Avanzata – IDA PRO

L'esercizio di oggi ha il fine di fare pratica con IDA PRO, un disassembler molto usato per praticare del Reverse Engineering su un malware.

Andiamo quindi ad utilizzarlo per il nostro file malevolo:

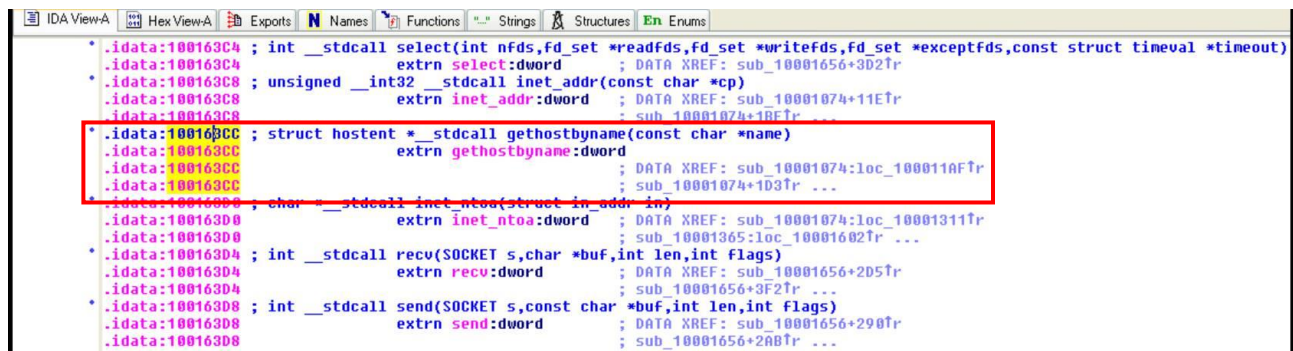
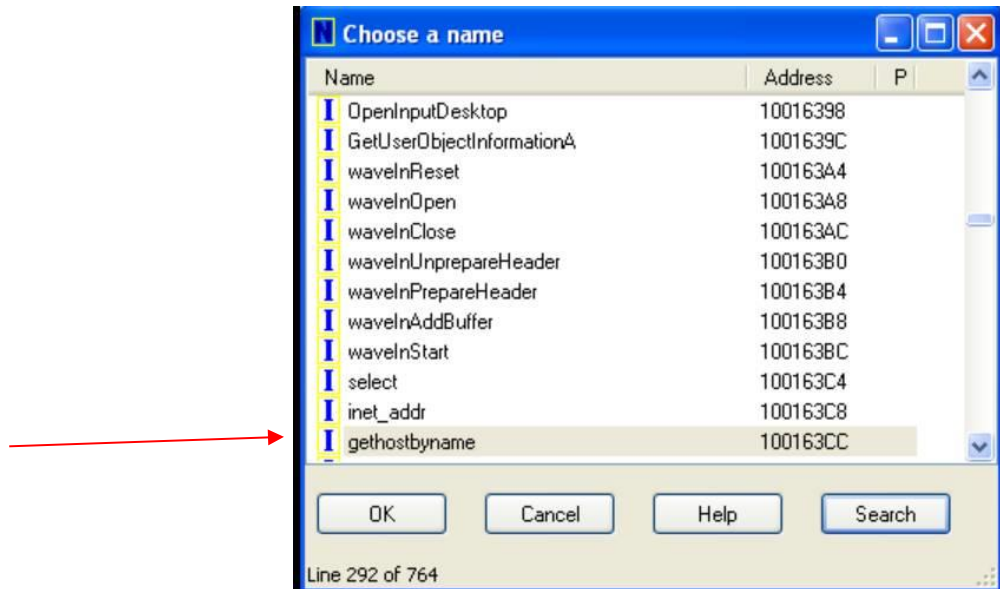
## FUNZIONE DLLMAIN

Per fare ciò, andiamo ad inserire in 'jump to functions' inseriamo la stringa DLLMain per individuare l'indirizzo della funzione:



## FUNZIONE GetHostByName

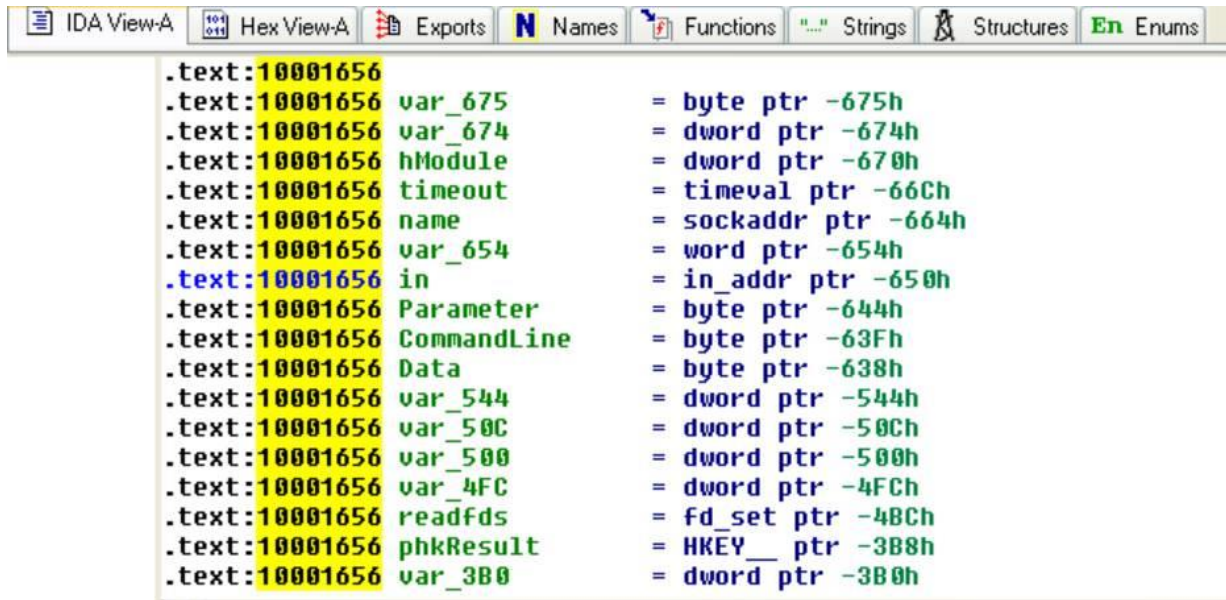
Con lo stesso procedimento 'jump to name', andiamo a cercare l'import 'gethostbyname':



La funzione viene utilizzata per ricavare informazioni riguardo la rete di un determinato host.

## VARIABILI LOCALI DELLA FUNZIONE 0x10001656

Con 'jump to address' possiamo andare a cercare il nostro indirizzo dato dalla traccia.



```
.text:10001656  
.text:10001656 var_675 = byte ptr -675h  
.text:10001656 var_674 = dword ptr -674h  
.text:10001656 hModule = dword ptr -670h  
.text:10001656 timeout = timeval ptr -66Ch  
.text:10001656 name = sockaddr ptr -664h  
.text:10001656 var_654 = word ptr -654h  
.text:10001656 in = in_addr ptr -650h  
.text:10001656 Parameter = byte ptr -644h  
.text:10001656 CommandLine = byte ptr -63Fh  
.text:10001656 Data = byte ptr -638h  
.text:10001656 var_544 = dword ptr -544h  
.text:10001656 var_50C = dword ptr -50Ch  
.text:10001656 var_500 = dword ptr -500h  
.text:10001656 var_4FC = dword ptr -4FCh  
.text:10001656 readfds = fd_set ptr -48Ch  
.text:10001656 phkResult = HKEY__ ptr -3B8h  
.text:10001656 var_3B0 = dword ptr -3B0h
```

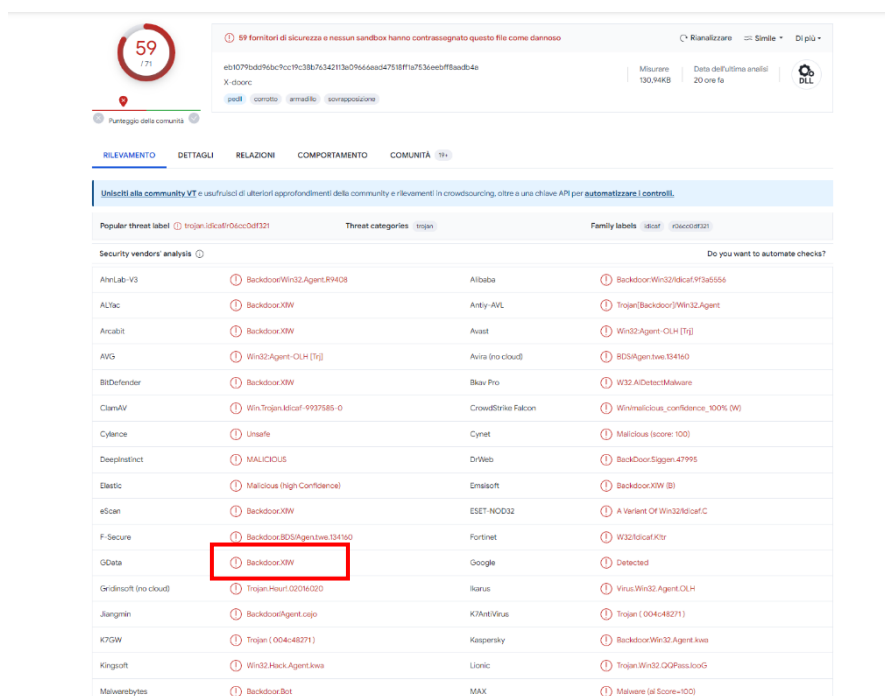


```
.text:10001656 var_1A4 = dword ptr -1A4h  
.text:10001656 var_194 = dword ptr -194h  
.text:10001656 WSADATA = WSADATA ptr -190h  
.text:10001656 arg_0 = dword ptr 4
```

Le variabili locali trovate risultano 20, più 1 parametro 'arg\_0'.

## CONSIDERAZIONI MACRO LIVELLO

Possiamo fare delle considerazioni utilizzando VirusTotal. Per fare ciò usiamo 'md5deep' per elaborare il codice hash del nostro file per poi inserirlo su VirusTotal.



59

59 fornitori di sicurezza e nessun sandbox hanno contrassegnato questo file come dannoso

eb070dd76bc9cc9c38b7634213e0f66e6a47518f1a763ee0ff8a0d4e

X-dboac

Misurare 130,94KB

Data dell'ultima analisi 20 ore fa

Analizza Simile Di più

pedi comito amello sovrapposizione

Punteggio della comunità

RILEVAMENTO DETTAGLI RELAZIONI COMPORTAMENTO COMUNITÀ (1)

Unisciti alla community VT e usufruisci di ulteriori approfondimenti della community e rilevamenti in crowdsourcing, oltre a una chiave API per automatizzare i controlli.

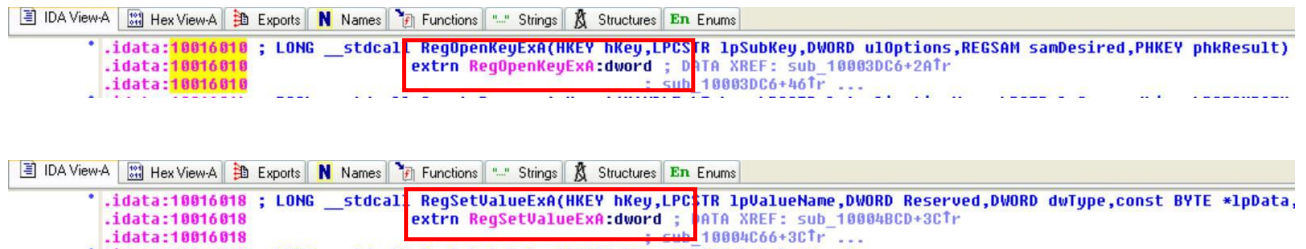
Popular threat label Trojan:Win32/Agent.R9408 Threat categories Trojan Family labels Backdoor:Win32/Agent.R9408

Security vendors' analysis Do you want to automate checks?

AhnLab-V3	Backdoor:Win32/Agent.R9408	Alibaba	Backdoor:Win32/Agent.R9408
ALYac	Backdoor:Win32/Agent.R9408	Antiy-AVL	Trojan(Backdoor)/Win32/Agent
ArcaBit	Backdoor:Win32/Agent.R9408	Avast	Win32/Agent-OLH [Trj]
AVG	Win32/Agent-OLH [Trj]	Avira (no cloud)	SDS/Agent.twe.154160
BitDefender	Backdoor:Win32/Agent.R9408	Bkav Pro	W32/AIDetect/Malware
ClamAV	Win32/Agent-OLH [Trj]	CrowdStrike Falcon	Win32/Agent-OLH [Trj]
Cybereason	Backdoor:Win32/Agent.R9408	Cyren	Malicious (score: 100)
DeepInstinct	Backdoor:Win32/Agent.R9408	DigWeb	Backdoor:Win32/Agent.R9408
Elastic	Backdoor:Win32/Agent.R9408	Emisoft	Backdoor:Win32/Agent.R9408
eScan	Backdoor:Win32/Agent.R9408	ESET-NOD32	A Variant Of Win32/Agent.R9408
F-Secure	Backdoor:Win32/Agent.R9408	Fortinet	W32/Agent.R9408
GData	Backdoor:Win32/Agent.R9408	Google	Detected
GridinSoft (no cloud)	Trojan:Win32/Agent.R9408	Ikarus	Virus:Win32/Agent.R9408
Jiangmin	Backdoor:Win32/Agent.R9408	K7AntiVirus	Trojan (00448271)
K7GW	Trojan (00448271)	Kaspersky	Backdoor:Win32/Agent.R9408
Kingsoft	Win32/Hack.Agent.R9408	Lionic	Trojan:Win32/Agent.R9408
Malwarebytes	Backdoor:Win32/Agent.R9408	MAX	Malware (af Score=100)

Da qui possiamo intuire che lo scopo del malware è quello di creare una backdoor per un potenziale attaccante.

Inoltre, andando a spulciare il codice con IDA, possiamo notare le funzioni RegOpenKeyEx (che permette di aprire una chiave di registro al fine di modificarla) e RegSetValueEx (che permette invece di aggiungere un nuovo valore all'interno del registro e di settare i rispettivi dati). Questo è segno che si cerca di ottenere la persistenza e quindi potrebbe confermare l'ipotesi di una backdoor.



```
IDA View-A Hex View-A Exports Names Functions Strings Structures Enums
* .idata:10016010 ; LONG __stdcall RegOpenKeyEx(HKEY hKey,LPCSTR lpSubKey,DWORD uOptions,REGSAM samDesired,PHKEY phkResult)
* .idata:10016010 extrn RegOpenKeyExA:dword ; DATA XREF: sub_10003DC6+2A1r
* .idata:10016010 * sub_10003DC6+461r ...

IDA View-A Hex View-A Exports Names Functions Strings Structures Enums
* .idata:10016018 ; LONG __stdcall RegSetValueEx(HKEY hKey,LPCSTR lpValueName,DWORD Reserved,DWORD dwType,const BYTE *lpData,
* .idata:10016018 extrn RegSetValueExA:dword ; DATA XREF: sub_10004BCD+3C1r
* .idata:10016018 * sub_10004C66+3C1r ...
```