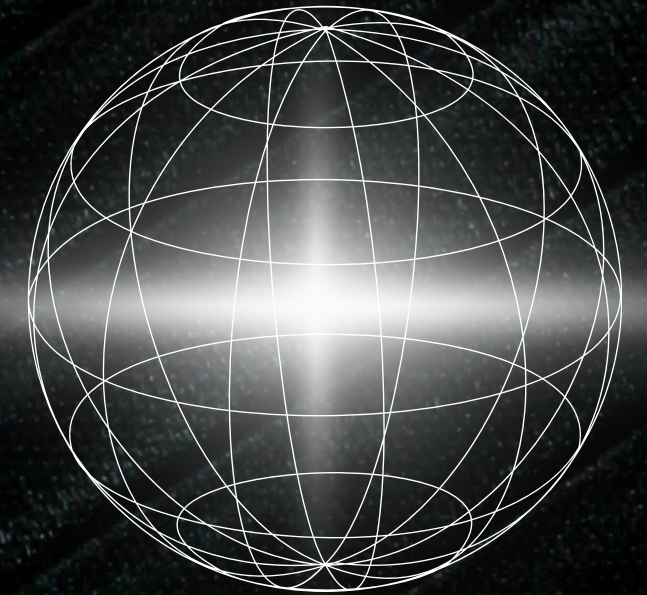


# Report sull'Analisi Statica del Malware "Malware\_U3\_W3\_L2"



Traccia

## Traccia:

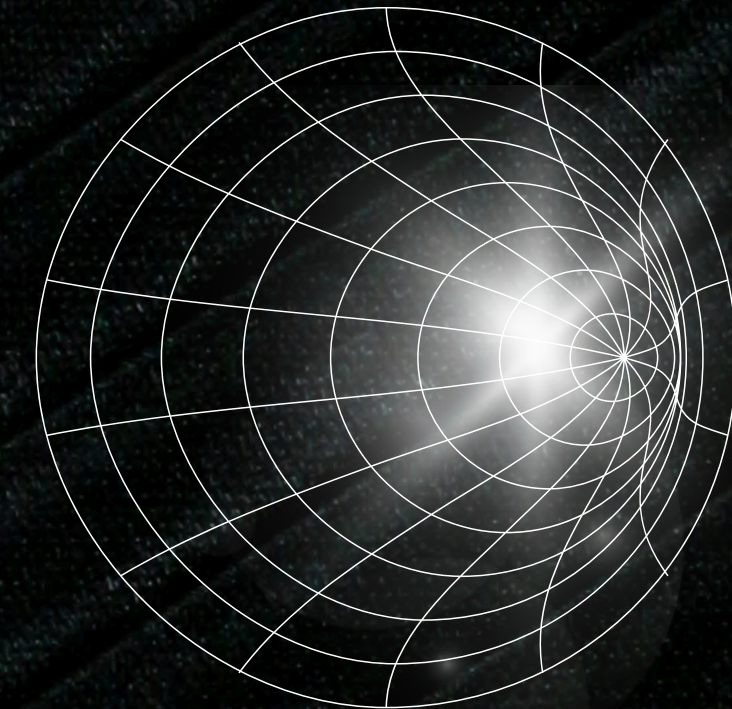
Lo scopo dell'esercizio di oggi è di acquisire esperienza con IDA, un tool fondamentale per l'analisi statica.

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)



# Introduzione

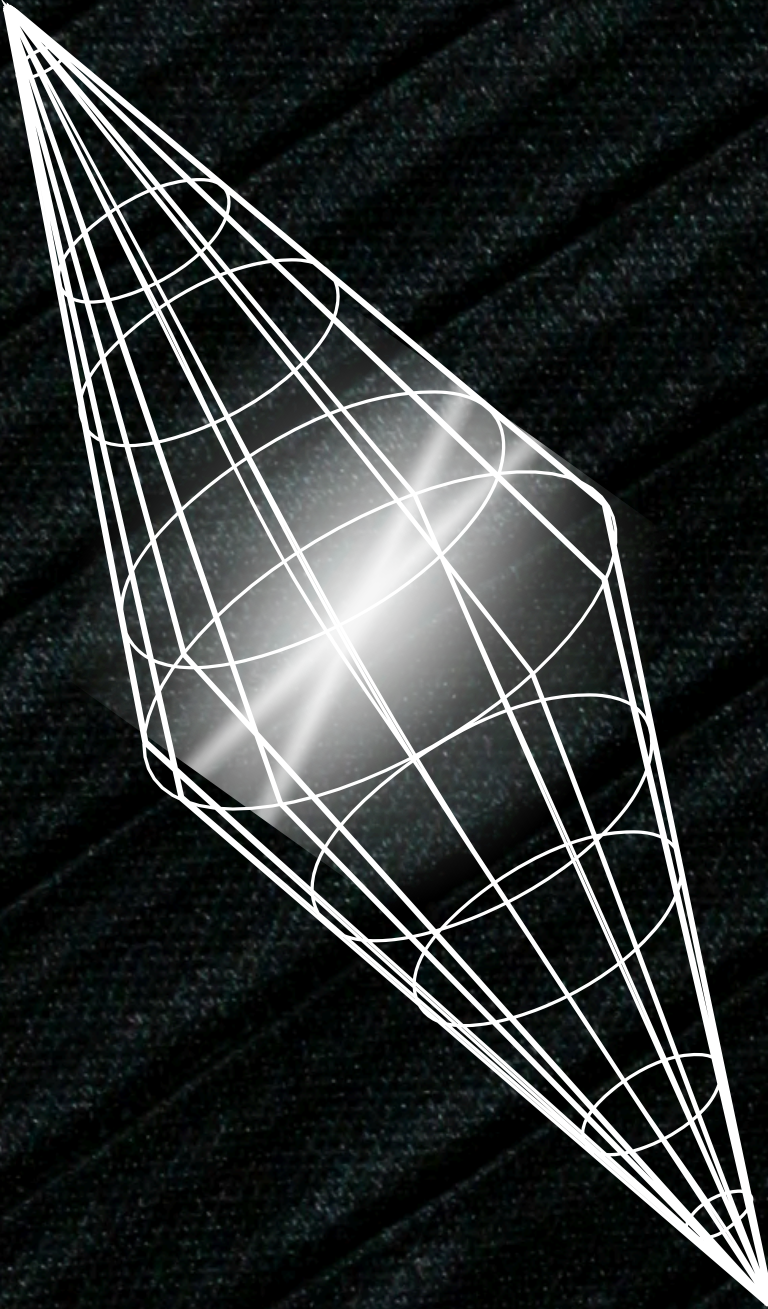


L'analisi statica è una fase fondamentale per comprendere il funzionamento e il comportamento di un malware. Utilizzando IDA Pro, uno strumento avanzato di disassembly, analizziamo il malware denominato "Malware\_U3\_W3\_L2" per rispondere a specifiche domande e fornire una visione dettagliata delle sue funzionalità.



# 01 Individuazione dell'indirizzo della funzione DLLMain:

La funzione DLLMain è individuata attraverso la sua definizione e il suo indirizzo di memoria nel disassembly.  
Come evidenziato nell'immagine:  
L'indirizzo esadecimale della funzione DLLMain è 0x1000D02E.



```
; 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

mov     eax, [esp+fdwReason]
dec     eax
jnz     loc_1000D107

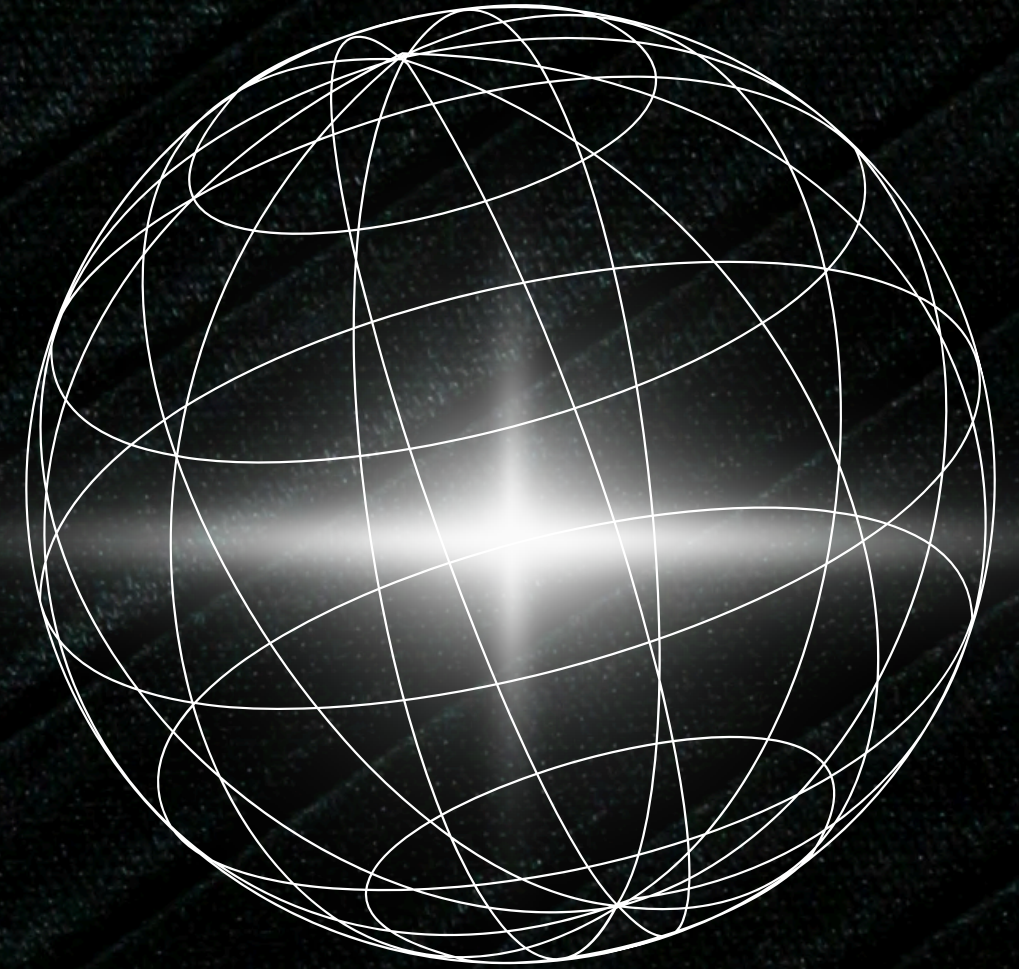
(-59,-141) | (586,349) | 0000C42E | 1000D02E: DllMain(x,x,x)
```



## 02 Funzione gethostbyname nella scheda "imports":

Consultando la scheda "imports" di IDA Pro, possiamo trovare la funzione gethostbyname. Questa funzione fa parte della libreria di rete e viene utilizzata per risolvere un hostname in un indirizzo IP. L'indirizzo di importazione di questa funzione nel nostro malware è 0x100163CC. La funzione gethostbyname risolve un nome di dominio in un indirizzo IP, consentendo al malware di comunicare con server remoti utilizzando nomi di dominio.

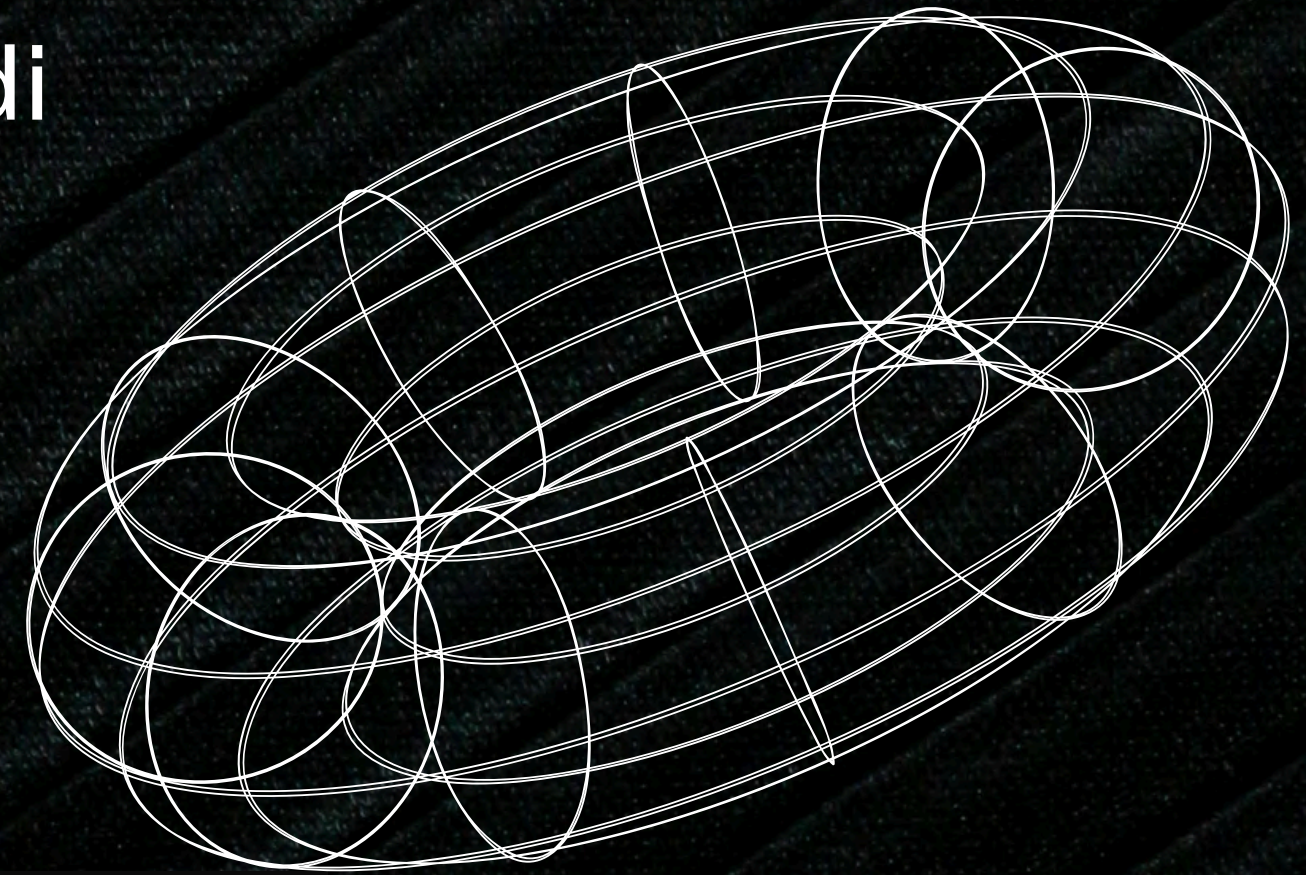
```
.idata:100163CC ; struct hostent *__stdcall gethostbyname(const char *name)
```





# 03 Variabili locali della funzione alla locazione di memoria 0x10001656:

Analizzando la funzione situata all'indirizzo 0x10001656, possiamo identificare numerose variabili locali definite nel contesto di questa funzione. Nel codice disassemblato, le variabili locali sono mostrate come segue:  
In totale, possiamo identificare 22 variabili



```
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= dword 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_50C= dword ptr -50Ch
var_500= dword ptr -500h
Buf2= byte ptr -4FCh
readfds= fd_set ptr -4BCh
phkResult= byte ptr -388h
var_380= dword ptr -380h
var_1A4= dword ptr -1A4h
var_194= dword ptr -194h
WSAData= WSADATA ptr -190h
```



## 04 Parametri della funzione alla locazione di memoria 0x10001656:

I parametri di una funzione sono identificabili osservando l'uso dei registri e degli offset relativi a EBP nel prologo della funzione. Alla locazione di memoria 0x10001656, il parametro passato alla funzione include:

```
arg_0 = dword ptr 4
```



## 05 Considerazioni Macro sul Malware:

Il malware "Malware\_U3\_W3\_L2" utilizza funzioni di rete come `gethostbyname` per risolvere nomi di host, suggerendo che potrebbe tentare di comunicare con server remoti. L'uso della funzione `CreateThread` indica che il malware crea thread separati per eseguire compiti in parallelo, aumentando la complessità del suo comportamento e migliorando la sua capacità di svolgere attività multiple contemporaneamente. La presenza di stringhe come "`http://`" e "`ftp://`" all'interno del codice suggerisce che il malware potrebbe tentare di scaricare o caricare dati da e verso server remoti. L'analisi dei salti condizionali e delle chiamate di funzione indica che il malware potrebbe contenere logiche di controllo del flusso complesse per evitare la rilevazione o l'analisi.



## 06 - Conclusione

L'analisi statica del malware "Malware\_U3\_W3\_L2" utilizzando IDA Pro ci ha permesso di identificare la struttura e le funzionalità chiave del malware. La funzione DLLMain è localizzata all'indirizzo 0x1000D02E, e la funzione di importazione gethostbyname all'indirizzo 0x100163CC. La funzione alla locazione di memoria 0x10001656 contiene 22 variabili locali e un parametro. Le considerazioni macro suggeriscono che il malware utilizza funzionalità di rete per comunicare con server remoti e impiega tecniche avanzate per eseguire compiti in parallelo e potenzialmente eludere la rilevazione.