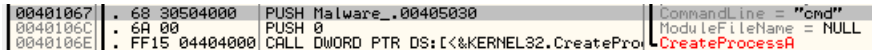
OllyDBG

In riferimento al malware: **Malware_U3_W3_L3,** rispondere ai seguenti quesiti utilizzando OllyDBG.

All'indirizzo 0040106E il Malware effettua una chiamata di funzione alla funzione CreateProcess. Qual è il valore del parametro CommandLine che viene passato allo stack?

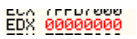
Il valore del parametro CommandLine, pushato nella funzione CreateProcess, è "**cmd**".



Inserite un breakpoint software all'indirizzo 004015A3. Qual è il valore del registro EDX? Eseguite a questo punto uno step-into. Indicate qual è ora il valore del registro EDX motivando la risposta. Che istruzione è stata eseguita?

Dopo aver inserito il breakpoint ed eseguito il run, nel registro si vede che il registro EDX ha valore di 00000A28

Eseguo poi uno step into e il valore che compare nel registro e di 00000000.



L'istruzione "XOR EDX, EDX" esegue un'operazione di XOR tra il registro EDX e stesso, azzerando il valore di EDX. Quando si esegue XOR tra un registro e se stesso, il risultato sarà sempre 0 perché i bit corrispondenti saranno uguali.

Inserite un secondo breakpoint all'indirizzo di memoria 004015AF. Qual è il valore del registro ECX? Eseguite uno step-into. Qual è ora il valore di ECX? Spiegate quale istruzione è stata eseguita.

Dopo aver inserito il secondo breakpoint ed eseguito il run per raggiungerlo, il registro di ECX è 0A280105

Eseguo poi uno step into e il valore che compare nel registro è 00000005

L'operazione AND viene eseguita confrontando bit a bit i due operandi, quindi i bit del registro di ECX e il valore in binario dell'esadecimale 0ff. Se entrambi i bit corrispondenti sono impostati a 1, il risultato sarà 1, altrimenti, il risultato sarà 0.

BONUS: spiegare a grandi linee il funzionamento del malware.

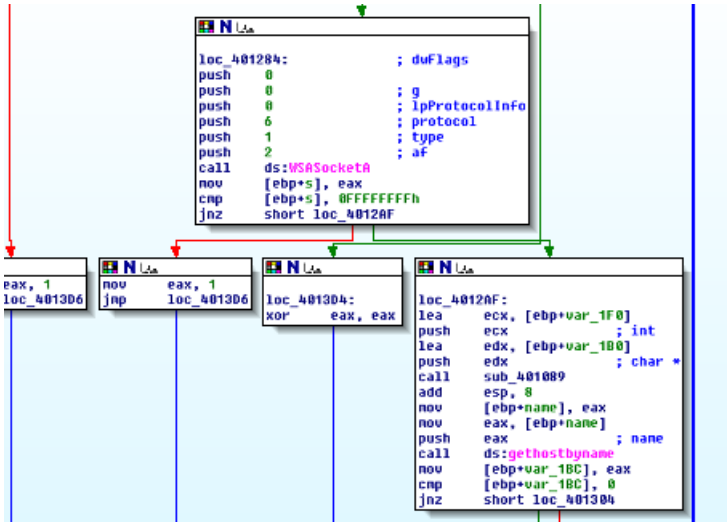
Per capire il funzionamento del malware, inizio con un'analisi statica basica, utilizzando CFF Explorer, per andare a vedere le librerie con le relative funzioni importate e le sezioni.

Le librerie importate sono due e sono Kernel32.dll e WS2_32.dll.

KERNEL32.dll	38	00004460	00000000	00000000	00004562	00004000
WS2_32.dll	7	000044FC	00000000	00000000	0000457E	0000409C

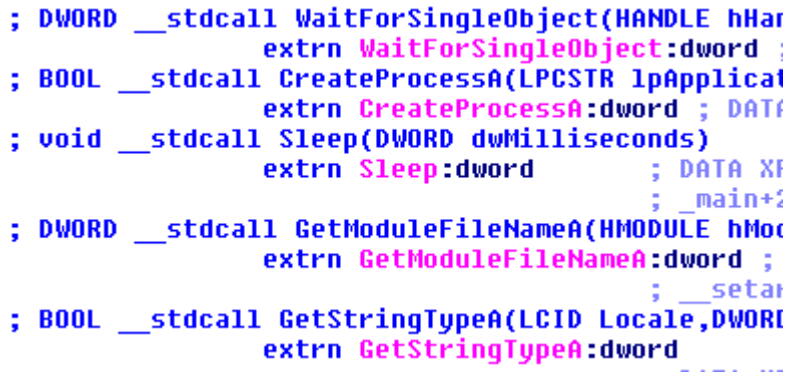
Mi sposto su IDA e dalla funzione main, vedo che il malware cerca di aprire una connessione socket, richiamando le funzioni di WS2_32.dll.

Il contenuto del codice è stato nascosto per rendere più difficile l'analisi, infatti spostandomi su IDA, su Hex View-A ci sono parti di codice binario vuote e scritte in modo casuale.



Dalle funzioni in basso, vengono richiamate funzioni dove vengono catturate varie informazioni alla macchina attaccata, come ad esempio getSting, GetmoduleFileName, GetCurrentProcess, TerminateProcess, GetVersion.

```
; LPSTR GetCommandLineA(void)
                extrn GetCommandLineA:dword ; DATA XREF:
; DWORD GetVersion(void)
                extrn GetVersion:dword  ; DATA XREF: sta
                                        ; Get current ve
                                        ; and informatio
; void __stdcall ExitProcess(UINT uExitCode)
                extrn ExitProcess:dword ; DATA XREF: _fa
                                        ; _doexit+91↑r
; BOOL __stdcall TerminateProcess(HANDLE hProcess,UINT u
                extrn TerminateProcess:dword ; DATA XREF
; HANDLE GetCurrentProcess(void)
                extrn GetCurrentProcess:dword ; DATA XRE
; LONG __stdcall UnhandledExceptionFilter(struct _EXCEPT
                extrn UnhandledExceptionFilter:dword
                                        ; DATA XREF: __X
; BOOL __stdcall FreeEnvironmentStringsA(LPSTR)
                extrn FreeEnvironmentStringsA:dword
                                        ; DATA XREF: ___
; BOOL __stdcall FreeEnvironmentStringsW(LPWSTR)
                extrn FreeEnvironmentStringsW:dword
```



Dando uno sguardo all'analisi effettuate e alle funzioni importate dal malware, potrebbe trattarsi di una backdoor che recupera informazioni del sistema attaccato, dopo aver instaurato la connessione.