Reverse Engineering

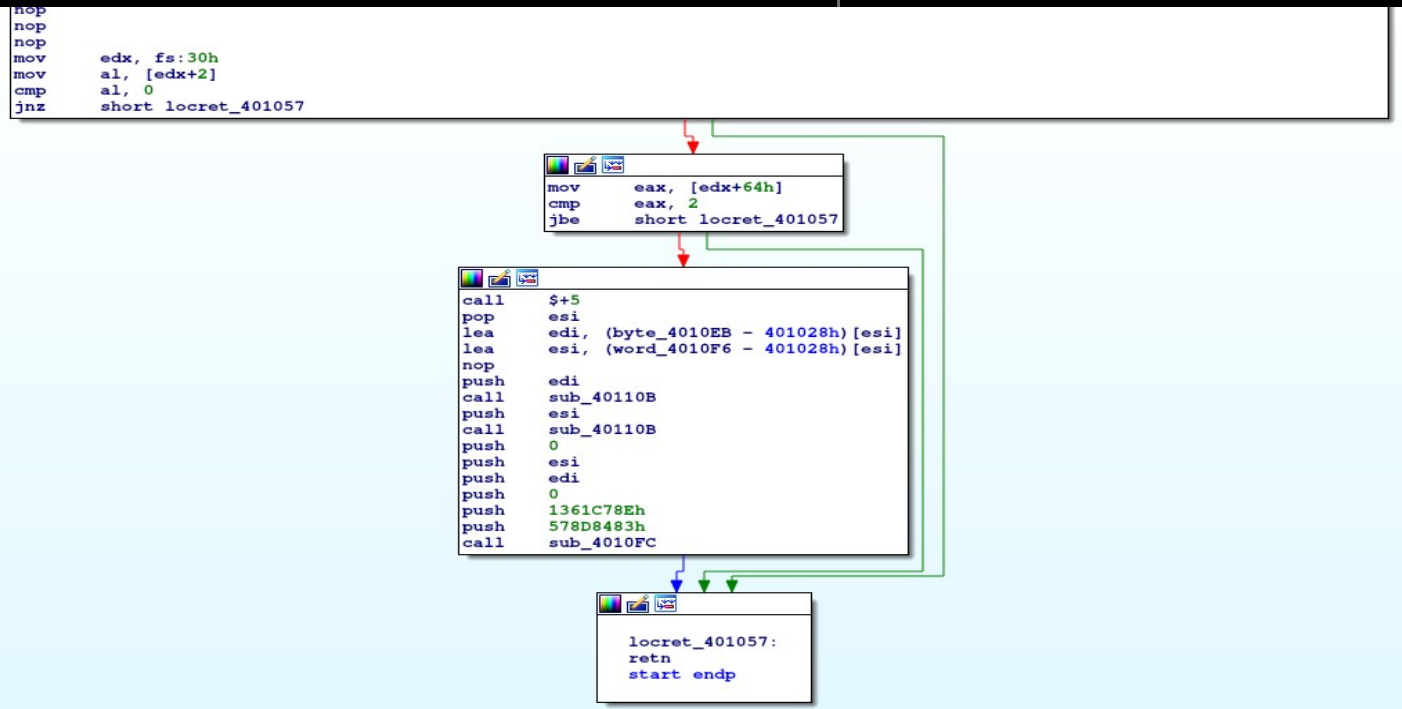
Audric De Pierrepont, Hadrien Cortade, Gautier Bielak

Sommaire

- Exercice 1
  - Raisonnement
  - Réponses
- Exercice 2
  - Raisonnement
  - Réponses

Exercice 1 : raisonnement

- Analyse graphe IDA



Exercice 1 : raisonnement

Min version	XP	XP SP2	2003/XP64	2003/XP64 SP1	Vista	Vista SP1	7	8 Pre RTM	8 Pre RTM	8.1 Update 1	10 TH2
Max version	XP SP1	XP SP3		2003/XP64 SP2		Vista SP2	7 SP1		8	10	
x86 offset offset:bitpos	Field Name										
0x0000	uint8_t InheritedAddressSpace										
0x0001	uint8_t ReadImageFileExecOptions										
0x0002	uint8_t BeingDebugged										
0x0003	uint8_t SpareBool			uint8_t BitField							
0x0003:0x00				uint8_t ImageUsesLargePages							
0x0003:0x01				uint8_t SpareBits	uint8_t IsProtectedProcess						
0x0003:0x02					uint8_t IsLegacyProcess					uint8_t IsImageDynamicallyRelocated	
0x0003:0x03					uint8_t IsImageDynamicallyRelocated					uint8_t SkipPatchingUser32Forwarders	
0x0003:0x04					uint8_t SpareBits	uint8_t SkipPatchingUser32Forwarders				uint8_t IsPackagedProcess	
0x0003:0x05						uint8_t SpareBits		uint8_t IsPackagedProcess		uint8_t IsAppContainer	
0x0003:0x06								uint8_t IsAppContainer		uint8_t IsProtectedProcessLight	
0x0003:0x07								uint8_t SpareBits			
0x0004	void * Mutant										
0x0008	void * ImageBaseAddress										
0x000C	struct _PEB_LDR_DATA * Ldr										
0x0010	struct _RTL_USER_PROCESS_PARAMETERS * ProcessParameters										
0x0014	void * SubSystemData										
0x0018	void * ProcessHeap										
0x001C	struct _RTL_CRITICAL_SECTION * FastPebLock										
0x0020	void * FastPebLockRoutine		void * SparePtr1	void * AtlThunkSListPtr							
0x0024	void * FastPebUnlockRoutine		void * SparePtr2		void * IFEOKey						
0x0028					unsigned long						

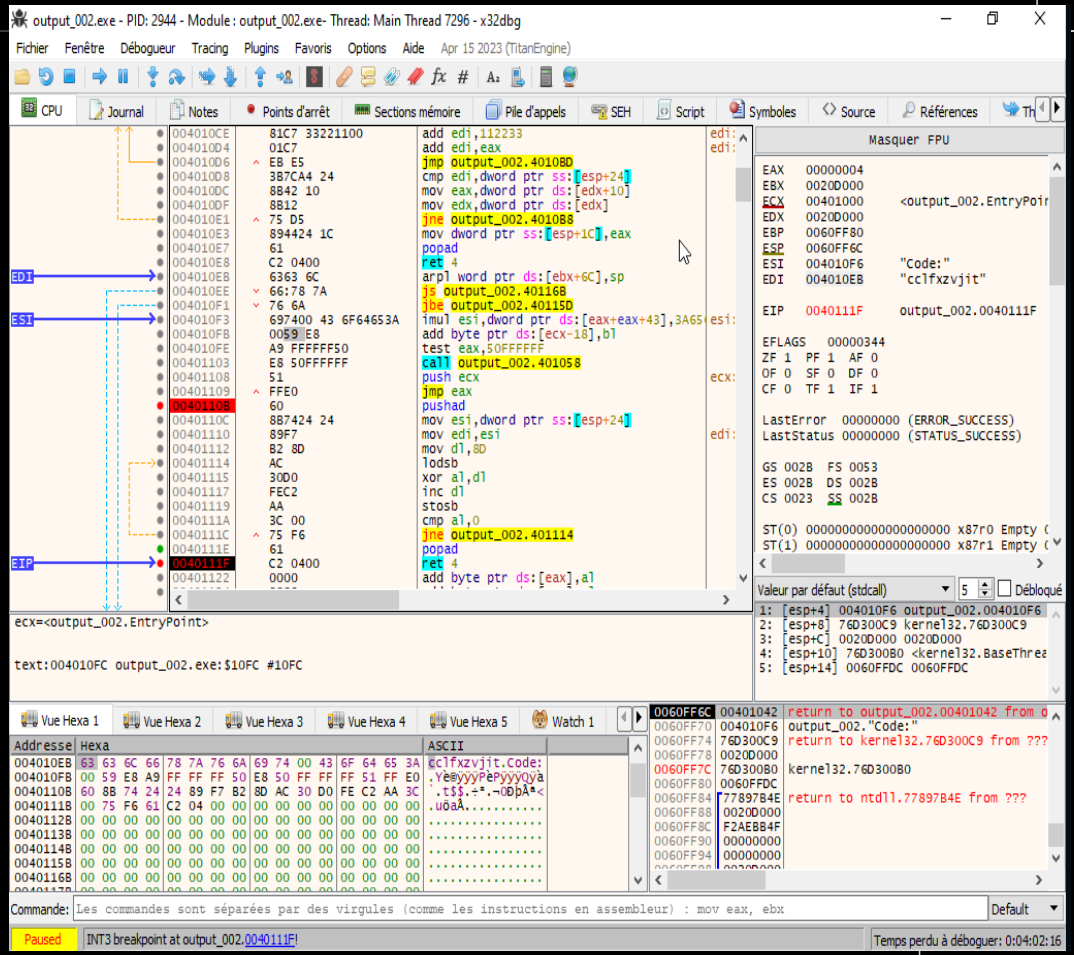
Exercice 1 : raisonnement

- Deux fonctions :
  - hash vu en cours
  - hash plus compliquée

```
sub_4010FC proc near
pop     ecx
call    sub_4010AB
push    eax
call    sub_401058
push    ecx
jmp     eax
sub_4010FC endp ; sp-analysis failed
```

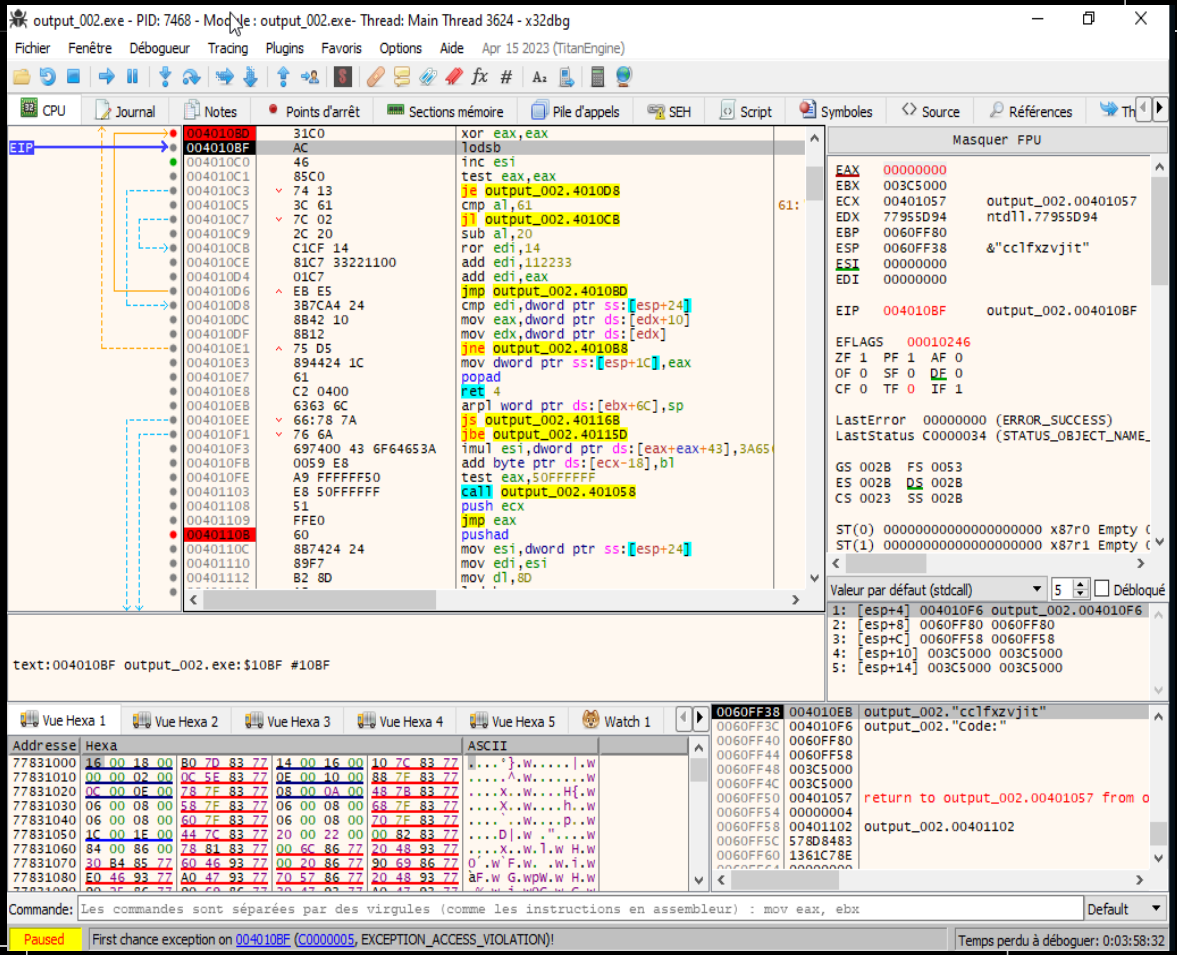
Exercice 1 : raisonnement

- Données importantes :
  - « cclfxzvjit Code : »
  - 0x578D8483
  - 0X1361C78E



Exercice 1 : raisonnement

- Problème encouru



Exercice 1 : réponses

1. Deux conditions :

le programme ne doit pas détecter qu'il est debuggé

- il faut minimum 2 processeurs

Exercice 1 : réponses

2. Structures :

PEB, LDR, TEB

- 3. Explication du shellcode

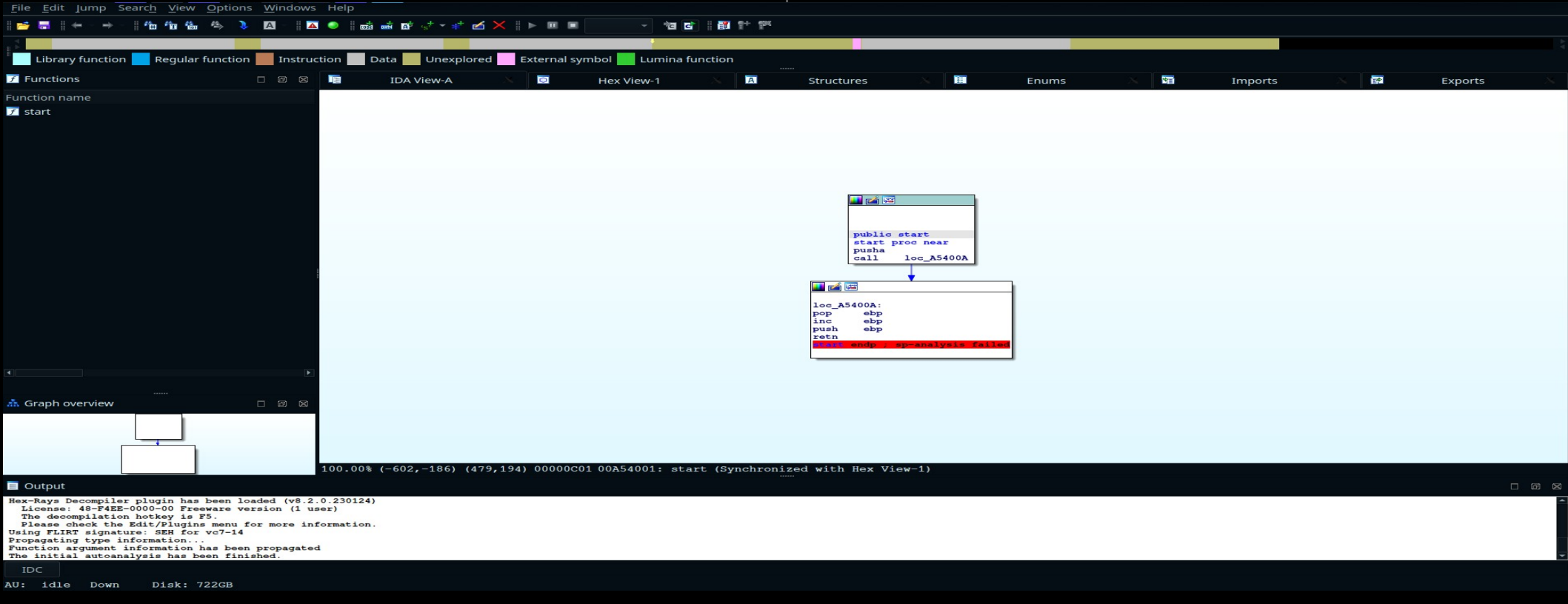
Exercice 1 : réponses

4. Hash de Kernel32.dll

Kernel32.DLL = D09F8780

Exercice 2 : raisonnement

- Analyse IDA



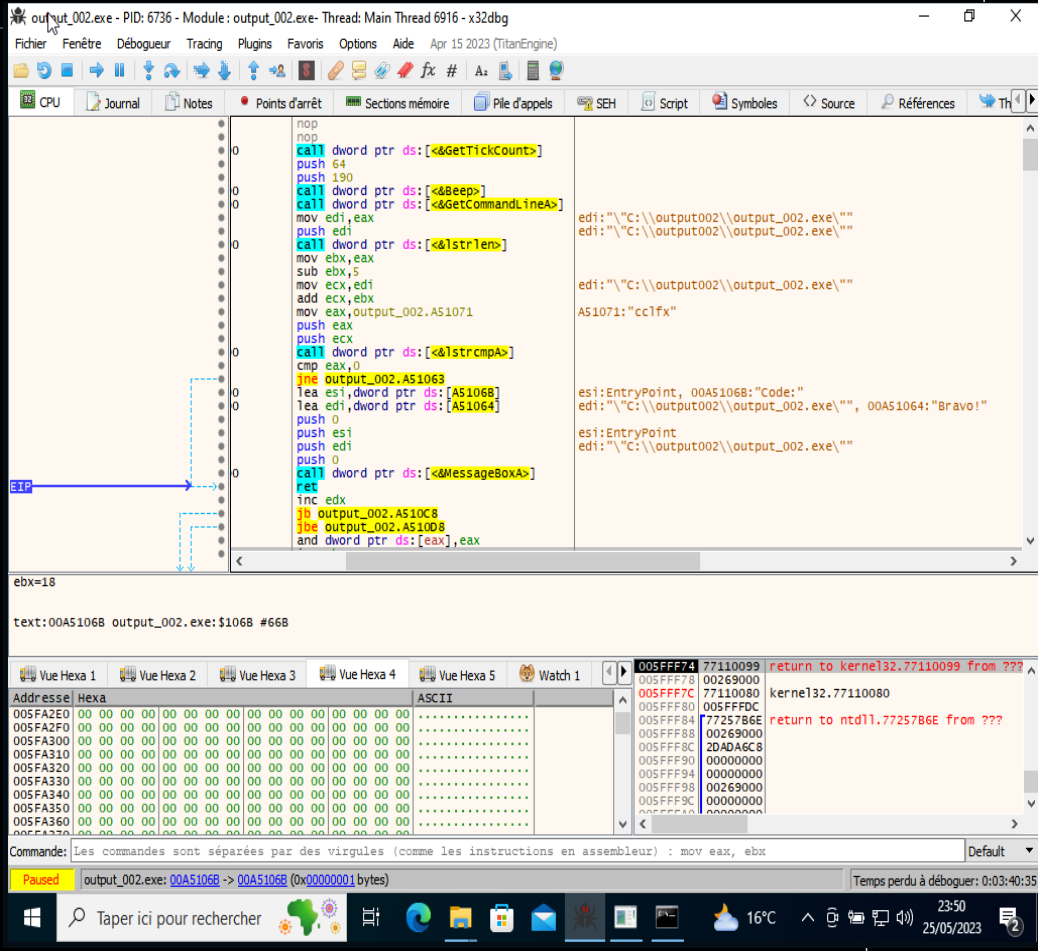
Exercice 2 : raisonnement

- Type de packing :

  ASPacking

Exercice 2 : raisonnement

Binaire dépaqué



Exercice 2 : réponses

Conditon :

Lancer le programme avec « cclfx » comme argument

Merci à vous