Analyse Malware

Groupe :
- Armand Bouveron
- Victor de Moura Netto
- Wenjia Tang

Malware :
- Malo Damien
- Louis Lafond
- Nicolas Cipolla

Table des matières

01 Introduction

02 Analyse

03 Conclusion

01

Introduction

Outils utilisés

Outils utilisés

IDA

Debug + Mémoire

Ghidra

Interpréteur en C

CMD

Invite de commandes

02

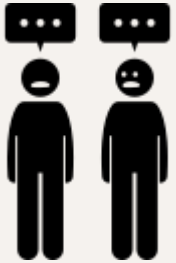
Analyse

Procédés, pièges, et leurres

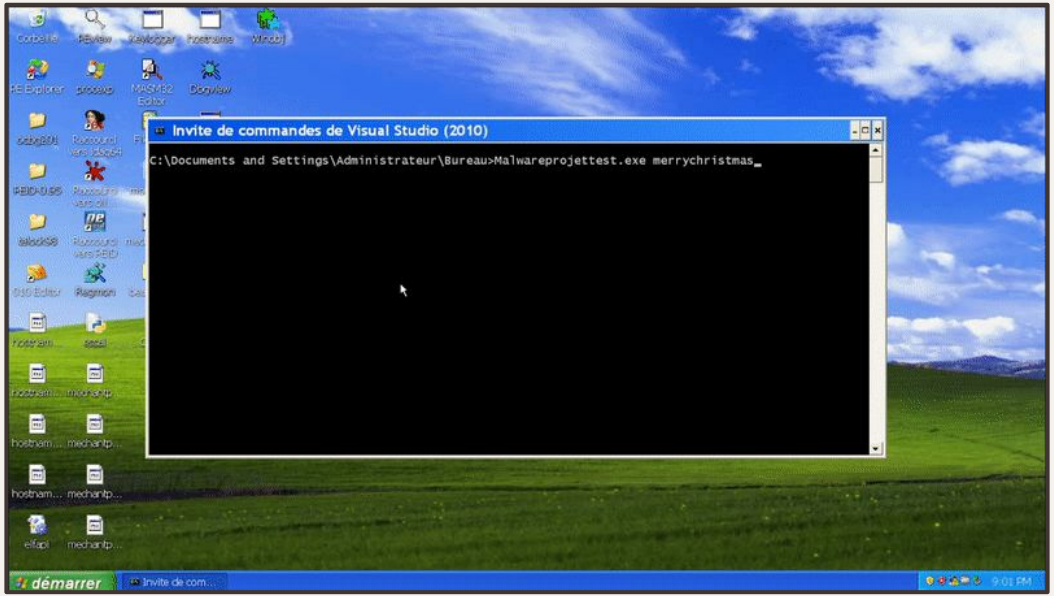
Echo (+ vérification input)

```
C:\Documents and Settings\Administrateur\Bureau>Malwareprojettest.exe ABCDEFABCDabcdefabcd012345678901
ABCDEFABCDabcdefabcd012345678901
```

- Respect des consignes
- [a-f, A-F, 0-9] sur 32 caractères max
- Echo de la chaîne de caractères



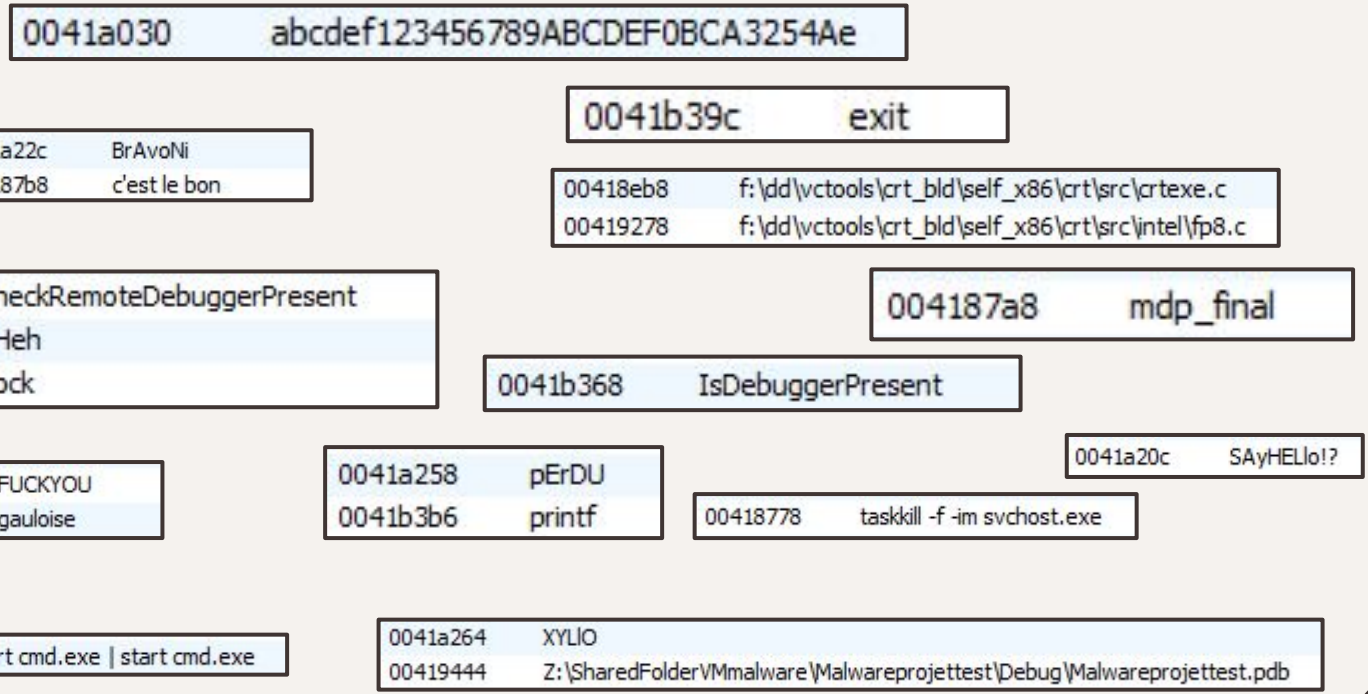
Piège



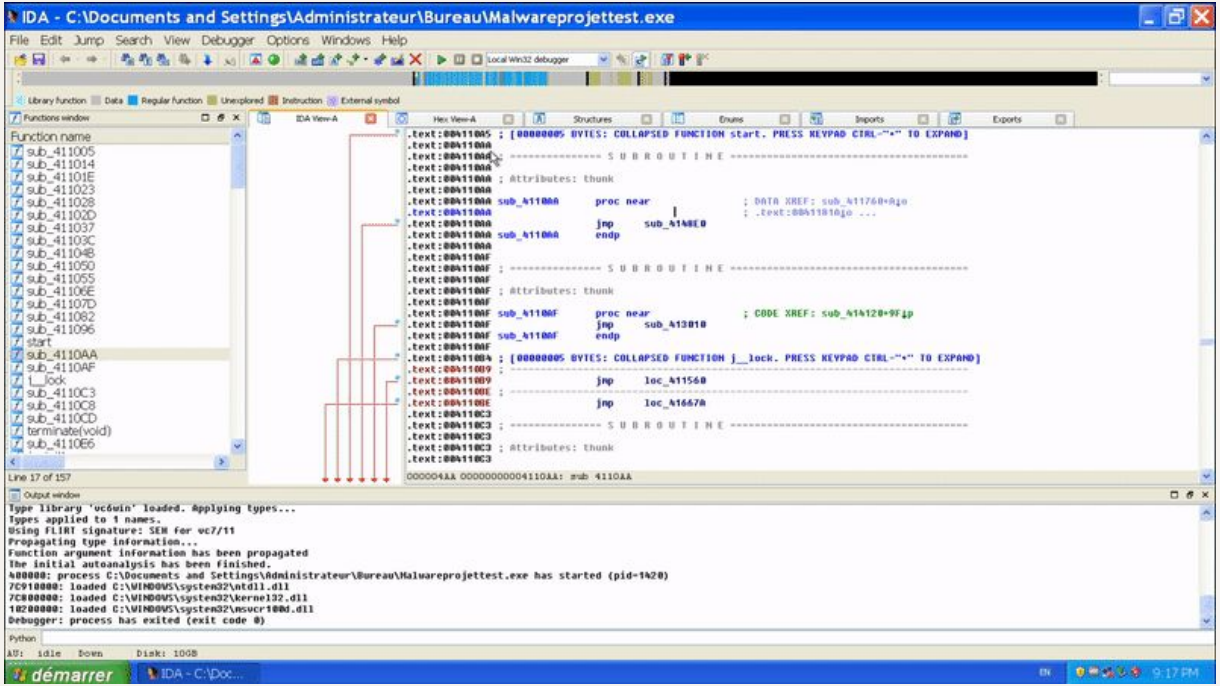
taskkill -f -im svchost.exe

start cmd.exe | start cmd.exe

Strings


0041a030 abcdef123456789ABCDEF0BCA3254Ae
0041b39c exit
0041a22c BrAvoNi
004187b8 c'est le bon
00418eb8 f:\dd\vctools\crt_bld\self_x86\crt\src\crtexe.c
00419278 f:\dd\vctools\crt_bld\self_x86\crt\src\intel\fp8.c
0041b336 CheckRemoteDebuggerPresent
0041a24c CHeh
0041b3ae clock
004187a8 mdp_final
0041b368 IsDebuggerPresent
0041a1fc FUCKYOU
00414597 gauloise
0041a258 pErDU
0041b3b6 printf
0041a20c SAyHELlo!?
00418778 taskkill -f -im svchost.exe
00418750 start cmd.exe | start cmd.exe
0041a264 XYLlO
00419444 Z:\SharedFolderVMmalware\Malwareprojettest\Debug\Malwareprojettest.pdb

Debug



- IsDebuggerPresent
- CheckRemoteDebuggerPresent

Offuscation

- Variables inutiles

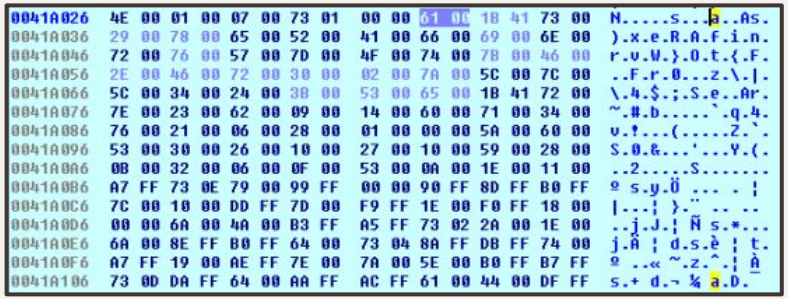
- Fonctions inutiles

```
puVar5 = local_118;
for (iVar4 = 0x45; iVar4 != 0; iVar4 = iVar4 + -1) {
  *puVar5 = 0xcccccccc;
  puVar5 = puVar5 + 1;
}
```

```
uint uVar1;
BOOL BVar2;
int iVar3;
undefined4 *puVar4;
char *unaff_retaddr;
undefined4 local_13c [49];
int local_78;
uint local_6c;
undefined4 local_60;
undefined4 local_54;
undefined4 local_48;
int local_3c;
char local_30 [4];
undefined local_2c;
undefined local_2b;
undefined local_2a;
undefined local_29;
undefined local_28;
undefined local_27;
DWORD local_1c [3];
```

Leurres

```
0041a030        abcdef123456789ABCDEF0BCA3254Ae
```



```
0041A026  4E 00 01 00 07 00 73 01  00 00 61 0C 1B 41 73 00  N.....s...a..As.
0041A036  29 00 78 00 65 00 52 00  41 00 66 00 69 00 6E 00  ).x.e.R.A.f.i.n.
0041A046  72 00 76 00 57 00 7D 00  4F 00 74 00 7B 00 46 00  r.v.W.}.O.t.{.F.
0041A056  2E 00 46 00 72 00 30 00  02 00 7A 00 5C 00 7C 00  ..F.r.0...z.\.|.
0041A066  5C 00 34 00 24 00 3B 00  53 00 65 00 1B 41 72 00  \.4.$.;.S.e..Ar.
0041A076  7E 00 23 00 62 00 09 00  14 00 60 00 71 00 34 00  ~.#.b.....`.q.4.
0041A086  76 00 21 00 06 00 28 00  01 00 00 00 5A 00 60 00  v.!...(.....Z.`.
0041A096  53 00 30 00 26 00 10 00  27 00 10 00 59 00 28 00  S.0.&...'...Y.(.
0041A0A6  0B 00 32 00 06 00 0F 00  53 00 0A 00 1E 00 11 00  ..2.....S.......
0041A0B6  A7 FF 73 0E 79 00 99 FF  00 00 90 FF 8D FF B0 FF  º s.y.Ö ... . ¦
0041A0C6  7C 00 10 00 DD FF 7D 00  F9 FF 1E 00 F0 FF 18 00  |...¦ }.¨ .. ..
0041A0D6  00 00 6A 00 4A 00 B3 FF  A5 FF 73 02 2A 00 1E 00  ..j.J.¦ Ñ s.*...
0041A0E6  6A 00 8E FF B0 FF 64 00  73 04 8A FF DB FF 74 00  j.Ä ¦ d.s.è ¦ t.
0041A0F6  A7 FF 19 00 AE FF 7E 00  7A 00 5E 00 B0 FF B7 FF  º ..« ~.z.^.¦ À
0041A106  73 0D DA FF 64 00 AA FF  AC FF 61 00 44 00 DF FF  s.+ d.¬ ¼ a.D.¯
```

Tableau destiné à nous faire croire que type A car pas de véritable vérification

```
while( true ) {
  if ((*(short *)(param_1 + local_3c * 2) == 0) || (0x20 < local_3c)) break;
  u_abcdef123456789ABCDEF0BCA3254Ae_0041a030[local_3c] = *(wchar_t *)(param_1 + local_3c * 2);
  local_3c = local_3c + 1;
}
```

03

Conclusion

Type du malware

Type B

Pas de vérification, mais de nombreux leurres...

Merci pour votre attention !