Déprotection semi-automatique de binaire

Metasm : celui qui fond dans la bouche et pas dans la main.

Alexandre Gazet
Yoann Guillot

Plan

Plan

Désassemblage

La référence : **IDA Pro**

- Excellent sur du code *clair* : binaire `MS`
- Inadapté sur un binaire protégé
  - Pas d'interprétation du code
  - Des hypothèses trop contraignantes

Des hypothèses

- Un appel retourne
- Deux instructions ne se superposent pas
- Les deux branches d'un saut conditionnel sont exécutables

Désassemblage

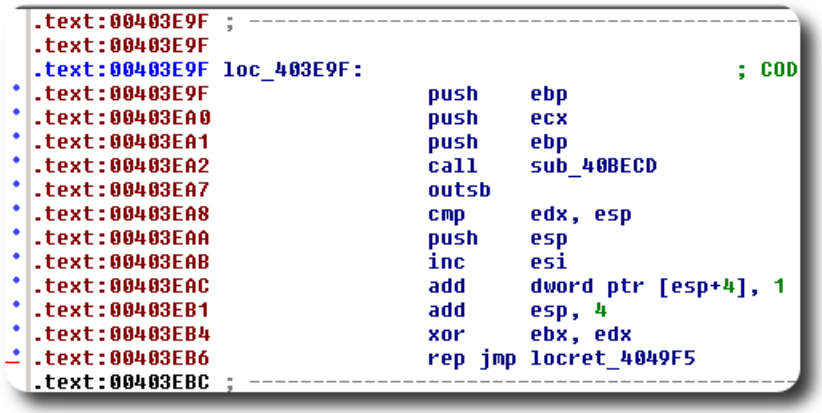
La référence : **IDA Pro**

- Excellent sur du code *clair* : binaire `MS`
- Inadapté sur un binaire protégé
  - Pas d'interprétation du code
  - Des hypothèses trop contraignantes

Des hypothèses

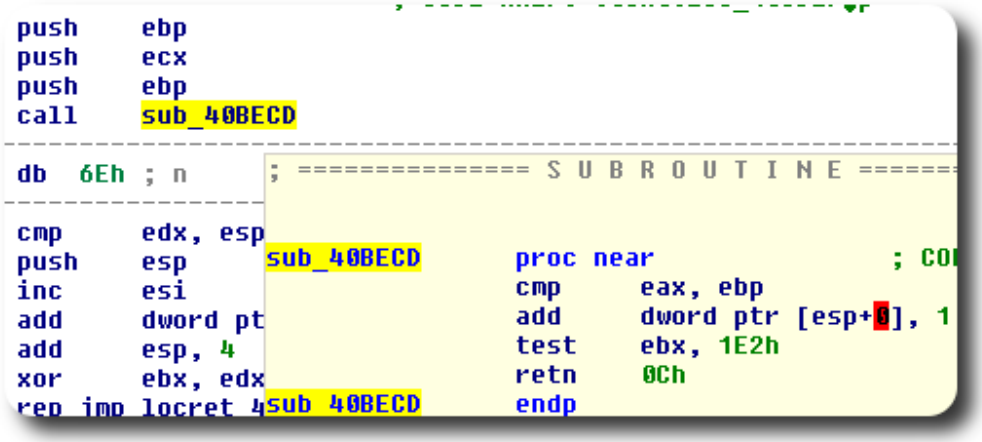
- Un appel retourne
- Deux instructions ne se superposent pas
- Les deux branches d'un saut conditionnel sont exécutables

Hypothèse : un appel retourne




```
.text:00403E9F loc_403E9F:      ; CODE XREF: .text:loc_40CDEF
.text:00403E9F                  push    ebp
.text:00403EA0                  push    ecx
.text:00403EA1                  push    ebp
.text:00403EA2                  call    sub_40BECD
.text:00403EA7                  outsb
.text:00403EA8                  cmp     edx, esp
.text:00403EAA                  push    esp
.text:00403EAB                  inc     esi
```

Mise en échec




```
.text:0040BECD sub_40BECD      proc near      ; CODE XREF: .text:00403EA2

.text:0040BECD                 cmp     eax, ebp
.text:0040BECF                 add     dword ptr [esp+0], 1
.text:0040BED4                 test    ebx, 1E2h
.text:0040BEDA                 retn    0Ch

.text:0040BEDA sub_40BECD      endp
```

Binding

Définition

Expression symbolique des effets d'une instruction ; à chaque instruction est associée sa sémantique.

```
a = di.instruction.args.map

res = Expression[[a[0], :&, mask], :+, [a[1], :&, mask]]

binding[:eflag_z] = Expression[[res, :&, mask], :==, 0]
binding[:eflag_s] = sign[res]
binding[:eflag_c] = Expression[res, :>, mask]
binding[:eflag_o] = Expression[[sign[a[0]], :==, sign[a[1]]],
            :'&&', [sign[a[0]], :'!=', sign[res]]]

binding[instr] ={ a[0] => ret }
```

Backtracking

Définition

Émulation symbolique par remontée du flot d'instructions.

Backtracking x dword ptr [esp] for 40bedah ret 0ch

1. backtrace `40becfh add dword ptr [esp+0], 1`
   `dword ptr [esp] => dword ptr [esp]+1`
2. backtrace up `40becdh->403ea2h dword ptr [esp]+1`
3. backtrace `403ea2h call loc_40becdh`
   `dword ptr [esp]+1` $\Rightarrow$ `403ea8h`
4. backtrace result : `403ea8h`

Metasm

Listing produit :

```
loc_403e9fh:
     push ebp                    ; @403e9fh  55
     push ecx                    ; @403ea0h  51
     push ebp                    ; @403ea1h  55
     call loc_40becdh            ; @403ea2h  e826800000  noreturn
db 6eh                           ; @403ea7h

// Xrefs: 40bedah
loc_403ea8h:
     cmp edx, esp                ; @403ea8h  39e2
     push esp                    ; @403eaah  54
[...]

// Xrefs: 403ea2h
loc_40becdh:
     cmp eax, ebp                ; @40becdh  39e8
     add dword ptr [esp+0], 1    ; @40becfh  8344240001
     test ebx, 1e2h              ; @40bed4h  f7c3e2010000
     ret 0ch                     ; @40bedah  c20c00  x:loc_403ea8h
```

Plan