

Test SHA-256

Localisation: 0040331B **Fonction:** ?

Type: Hash **Sévérité:** Moyenne

```

1 .text:00401F6F ; __ unwind { // sub_474F60
2 .text:00401F6F             lea     eax, [ebp+var_A]
3 .text:00401F72             mov     [esp+8], eax    ; int
4 .text:00401F76             mov     dword ptr [esp+4], offset
      aHelloWorld ; "Hello, World!"
5 .text:00401F7E             mov     dword ptr [esp], offset
      dword_565030 ; int
6 .text:00401F85             mov     [ebp+fctx.call_site], 2
7 .text:00401F8C             call    sub_448120
8 .text:00401F91             mov     dword ptr [esp], offset
      sub_4020A0 ; _onexit_t
9 .text:00401F98             call    sub_409C20
10 .text:00401F9D             lea     eax, [ebp+var_A+1]
11 .text:00401FA0             mov     [esp+8], eax    ; int
12 .text:00401FA4             mov     dword ptr [esp+4], offset
      aBea8e217036cb3 ; "bea8e217036cb3b738e207fe5d40266828bc196"...
13 .text:00401FAC             mov     dword ptr [esp], offset
      dword_565034 ; int
14 .text:00401FB3             mov     [ebp+fctx.call_site], 1
15 .text:00401FBA             call    sub_448120
16 .text:00401FBF             mov     dword ptr [esp], offset
      sub_401FF0 ; _onexit_t
17 .text:00401FC6             call    sub_409C20
18 .text:00401FCB             lea     eax, [ebp+fctx]
19 .text:00401FCE             mov     [esp], eax    ; lpfctx
20 .text:00401FD1             call    _Unwind_SjLj_Unregister
21 .text:00401FD6             leave
22 .text:00401FD7             retn

```

Analyse

Dans la fonction ci-dessus, le programme stocke la valeur

bea8e217036cb3b738e207fe5d40266828bc1969fd8538d533ea39f4e40ffc8f

qui semble être un digest (SHA-256) en `dword_565034`.

```

1 .text:00403770 loc_403770:
2 .text:00403770             cmp     ecx, ecx
3 .text:00403772             repe   cmpsb
4 .text:00403774             jnz    loc_403333
5 .text:0040377A             mov     eax, [ebp+var_24]
6 .text:0040377D             mov     edi, [eax-0Ch]
7 .text:00403780             lea     edx, [eax-0Ch]
8 .text:00403783             test   edi, edi
9 .text:00403785             jnz    loc_403996

```

Analyse

Ensuite, le programme compare le SHA-256 de l'entrée utilisateur avec la valeur de hash. Si les deux sont identiques, le programme saute à la fonction située en `loc_403996`

```

1 loc_403996:
2 .text:00403996          mov     esi, [edx+8]
3 .text:00403999          test    esi, esi
4 .text:0040399B          js      short loc_4039B5
5 .text:0040399D          lea     eax, [ebp+var_24]
6 .text:004039A0          mov     [esp], eax
7 .text:004039A3          mov     [ebp+fctx.call_site], 25h ; '%'
8 .text:004039AD          call    sub_445EE0
9 .text:004039B2          mov     eax, [ebp+var_24]
10 .text:004039B5
11 .text:004039B5 loc_4039B5:
12 .text:004039B5          movzx  eax, byte ptr [eax]
13 .text:004039B8          mov     dword ptr [esp], offset Format
   ; "Byte 0: %02x\n"
14 .text:004039BF          mov     [ebp+fctx.call_site], 25h ; '%'
15 .text:004039C9          mov     [esp+4], eax
16 .text:004039CD          call    printf
17 .text:004039D2          jmp    loc_40378B

```

Analyse

La fonction loc_4039B5 permet d'afficher le premier octet de l'entrée.

Screenshot

```

.text:004032C4          mov     dword ptr [esp+4], offset dword_565030
.text:004032CC          mov     [esp], eax
.text:004032CF          mov     [ebp+fctx.call_site], 26h ; '&'
call    sub_402150
mov     [ebp+var_30], 7
mov     edi, ds:dword 565034
sub    esp, 4
mov     [ebp+var_34], 3
mov     esi, [ebp+var_30]
mov     edx, [ebp+var_30]
mov     ecx, [ebp+var_34]
mov     eax, [ebp+var_34]
mov     [ebp+var_38], offset unk_47DFC8
imul   edx, esi
mov     esi, [ebp+var_28]
mov     [ebp+var_3C], offset unk_47DFC8
imul   eax, ecx
mov     ecx, [esi-0Ch]
cmp    edx, eax
setne  al
cmp    ecx, [edi-0Ch]
mov     byte ptr [ebp+var_C+3], al
mov     [ebp+var_40], 0
jz     loc_403770
; CODE XREF: sub_403230+544+j
; sub_403230+5F51j
loc_403333:
mov     eax, [ebp+var_24]
mov     eax, [eax+5Ch]

```

Analyse

Après avoir identifié ce potentiel comportement, nous avons tenté de le produire. Pour cela nous avons cherché le hash directement dans l'executable afin de le modifier afin de le remplacer par un hash de notre choix : le SHA-256 de 1234

03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4

Screenshot

```

010480 01000374 03725f08 0100c98c 05720000 0c696267 05085f73 2209a6c0 002f9470 3f520507 09737465 72456c61 73736373
510528 00000000 53637265 656e4065 6c746572 00770000 433a5c55 73657273 56c6b873 5c417070 44617461 5c46f63 616c5c54
510576 656f0795 61737469 686f2e74 78740658 4c4f475d 5b606169 6502066 6f706566 00706168 626f2065 742d6cf 60627269
510624 682f6672 00000000 00000000 00000000 00000000 00000000 00000000 48656c6c 6f2c2057 6f726c64 21000000
510672 62656138 65323137 30333663 62336237 33386532 30376665 35643430 32363638 3236263 31393639 66643835 33386435
510720 33336561 33396634 65343066 66633866 00000000 62617369 635f7374 72696e67 3a3a5f53 5f636f6e 73747275 63742046
510768 554c4c20 666f7420 76616c69 64000000 58444542 55475d20 48617368 2076616c 69646174 696f6e3a 20504153 53454420
510816 7c204368 65636873 75603a20 30784445 41444245 45460042 79746520 3032780a 00646563 6f790075 6e757365
510864 64006661 68655f64 6174615f 31006661 68655f64 6174615f 3200606f 72655f66 616b655f 64617461 00666966 616c5f64

```

Screenshot

```

J40710 0360013c 0113f763 080f1717 16f18038 50660103 0c3b2060 0f760501 00700000 02000000 48656c6c 6f2c2057 21000000
510624 682f6672 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
510672 30336163 36373432 31366633 65313563 37363165 65323535 66303637 39353336 32336338 62333838 62343435
510720 39651313 6639738 64376533 34366634 00000000 62617369 635f7374 72696e67 3a3a5f53 5f636f6e 73747275 63742046
510768 554c4c20 666f7420 76616c69 64000000 58444542 55475d20 48617368 2076616c 69646174 696f6e3a 20504153 53454420

```

Analyse

Cependant, en effectuant `m.exe 1234` le comportement reste inchangé. Deux hypothèses sont donc à considérer :

- le programme effectue un test d'intégrité sur son contenu ;
- le hash identifié n'impacte en aucun cas la fonction legitimate `echo`.