PUSH EBP
MOV EBP, ESP
MOV EAX, dword ptr [EBP+ param1]
MOV ECX, dword ptr [EBP+ param2]
SUB ESP, 0x2C
PUSH EBX
PUSH ESI
PUSH EDI
LEA EDI, [EAX + ECX*0x1]
LEA EDX, [EDI - +0x18] → 488fe8

LAB_004779f6:
CMP EAX, EDX → eax = 400000, edx = 4088fe8, c-flag = 1
JC *LAB_004773d0*

LAB_00477a0a:
XOR EAX, EAX

LAB_004774f8:
POP EDI
POP ESI
POP EBX
LEAVE
RET

LAB_004773d0:
MOV ECX, dword ptr [EAX]
TEST ECX, ECX
JZ *LAB_004776c3*
MOV ESI, ECX
XOR ESI, 0x88bbdd8d → 882b87c0
CMP dword ptr [EAX+ 0x4], ESI → [400004] = 00000003
JNZ *LAB_004776c3*
XOR ECX, 0xddbca2b2
CMP dword ptr [EAX+ 0x8], ECX
JZ *LAB_004777a2*

LAB_004776c3:
INC EAX

JMP *LAB_004779f6*

LAB_004777a2:
MOV EDX, dword ptr [EAX]
MOV ESI, dword ptr [EAX+ 0xc]
LEA ECX, [EAX + 0x14]
MOV EAX, dword ptr [EAX+ 0x10]
XOR EAX, EDX
XOR ESI, EDX
MOV dword ptr [EBP + local10], ECX
MOV dword ptr [EBP + param2], EDX
MOV dword ptr [EBP + param1], EAX
TEST ECX, ECX
JZ *LAB_00477a0a* → non preso
CMP ESI, EAX
JA *LAB_00477a0a* → non preso
ADD ECX, ESI
CMP ECX, EDI
JA *LAB_00477a0a* → non preso
MOV EDI, dword ptr [->KERNEL32.DLL::VirtualAlloc]
PUSH 0x4
PUSH 0x3000
PUSH ESI
PUSH 0x0
CALL EDI ;=> KERNEL32.DLL::VirtualAlloc
MOV EBX, EAX
XOR EDX, EDX
MOV dword ptr [EBP + local_gc], EBX
CMP EBX, EDX

JZ *LAB_00477a0a* → non preso: controllo che il ritorno di virtual alloc sia diverso da 0

MOV dword ptr [EBP + local_g8], EDX
MOV EAX, FS:[0x18]
MOV EAX, dword ptr [EAX + 0X30]
MOVZX EAX, byte ptr [EAX + 0x2]
MOV dword ptr [EBP + local_g8], EAX
CMP dword ptr [EBP + local_g8], EDX
JZ *LAB_00477948*

XOR EAX, EAX
MOV dword ptr [EBP + local30], 0x6e72656b
MOV dword ptr [EBP + local2c], 0x32336c65
MOV dword ptr [EBP + local28], 0x6c6c642e
LEA EDI, [EBP + -0x20]
STOSB ES:EDI
MOV dword ptr [EBP + local20], 0x6f6c6c41
MOV dword ptr [EBP + local1c], 0x6e6f4363
MOV dword ptr [EBP + local18], 0x656c6f73
LEA EDI, [EDI + -0x10]
STOSB ES:EDI
LEA EAX, [EBP + -0x1c]
PUSH EAX
LEA EAX, [EBP + -0x2c]
PUSH EAX
CALL dword ptr [->KERNEL32.DLL::GetModuleHandleA]
PUSH EAX
CALL dword ptr [->KERNEL32.DLL::GetProcAddress]
PUSH ESI
PUSH EAX
PUSH EBX
CALL FID_conflict:_memcpy
ADD ESP, 0xc

LAB_00477073:
MOV EAX, EBX
JMP *LAB_004774f8*

LAB_00477948:
MOV EAX, dword ptr [EBP + param2]
ADD EAX, 0xa078c405
MOV dword ptr [EBP + param2], EAX
CMP ESI, EDX
JBE *LAB_004777eb*
MOV EAX, dword ptr [EBP + local10]
SUB EAX, EBX
MOV dword ptr [EBP + local10], EAX

LAB_0047749c:
MOV EAX, dword ptr [EBP + param2]

MOV CL, DL
AND CL, 0x1f
ROL EAX, CL → CL = 0: nessuna rotazione
MOV ECX, dword ptr [EBP + param2]
ROR ECX, 0x3
ADD EAX, ECX
MOV ECX, EDX
ROR ECX, 0xb
ADD ECX, 0x72462828
XOR EAX, ECX
MOV ECX, dword ptr [EBP + local10]
MOV dword ptr [EBP + param2], EAX
LEA EAX, [EDX + EBX * 0x1]
MOV CL, byte ptr [ECX + EAX * 0x1]
XOR CL, byte ptr [EBP + param2]
mov EBX, dword ptr [EBP, local_gc]
INC EDX
MOV byte ptr [EAX], CL
CMP EDX, ESI
JC *LAB_0047749c*

LAB_004777eb:
CMP ESI, dword ptr [EBP + param1]
JZ LAB_00477073
PUSH 0x4
PUSH 0x3000
PUSH dword ptr [EBP + param1]
PUSH 0x0
CALL EDI ;=> KERNEL32.DLL::VirtualAlloc
MOV EDI, dword ptr [KERNEL32.DLL::VirtualFree]
MOV dword ptr [EBP + param2], EAX
TEST EAX, EAX
JZ *LAB_004772d6*
CALL FUN_0046e870
PUSH ESI
PUSH EBX
LEA EAX, [EBP + 0x8]
PUSH EAX
PUSH dword ptr [EBP + param2]
CALL FUN_0046e940

```
ADD ESP, 0x10
PUSH 0x8000
PUSH 0x0
TEST EAX, EAX
JZ LAB_00477a07
PUSH dword ptr [EBP + param2]
CALL EDI ;=>KERNEL32.DLL::VirtualFree

    LAB_004772d6:
PUSH 0x8000
PUSH 0x0
PUSH EBX
CALL EDI ;=>KERNEL32.DLL::VirtualFree
JMP LAB_00477a0a

    LAB_00477a07:
PUSH EBX
CALL dword ptr [->KERNEL32.DLL::VirtualFree]
MOV EAX dword ptr [EBP + param2]
JMP LAB_004774f8
```