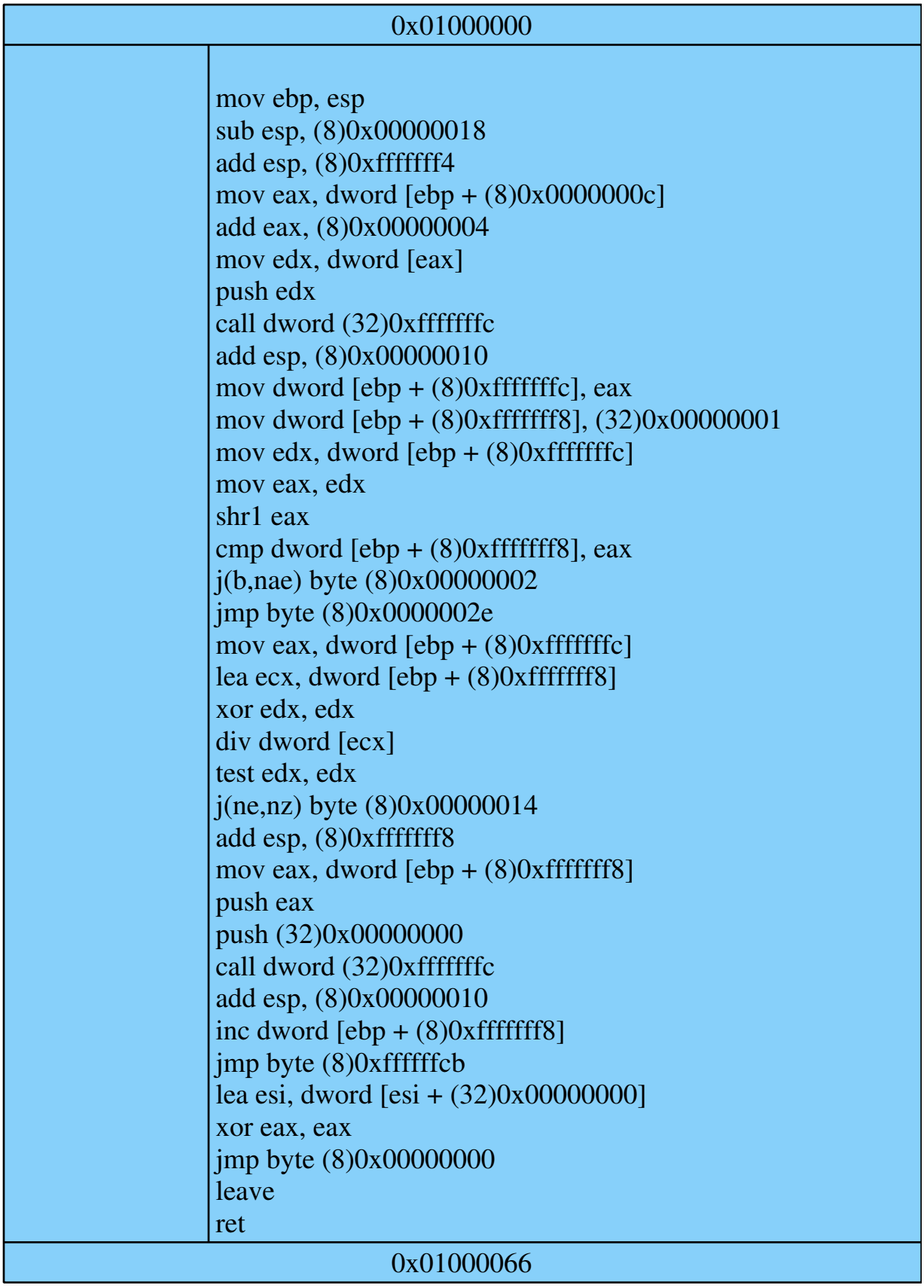
0x01000000
mov ebp, esp
sub esp, (8)0x00000018
add esp, (8)0xfffffff4
mov eax, dword [ebp + (8)0x0000000c]
add eax, (8)0x00000004
mov edx, dword [eax]
push edx
call dword (32)0xfffffffc
add esp, (8)0x00000010
mov dword [ebp + (8)0xfffffffc], eax
mov dword [ebp + (8)0xfffffff8], (32)0x00000001
mov edx, dword [ebp + (8)0xfffffffc]
mov eax, edx
shr1 eax
cmp dword [ebp + (8)0xfffffff8], eax
j(b,nae) byte (8)0x00000002
jmp byte (8)0x0000002e
mov eax, dword [ebp + (8)0xfffffffc]
lea ecx, dword [ebp + (8)0xfffffff8]
xor edx, edx
div dword [ecx]
test edx, edx
j(ne,nz) byte (8)0x00000014
add esp, (8)0xfffffff8
mov eax, dword [ebp + (8)0xfffffff8]
push eax
push (32)0x00000000
call dword (32)0xfffffffc
add esp, (8)0x00000010
inc dword [ebp + (8)0xfffffff8]
jmp byte (8)0xfffffffcb
lea esi, dword [esi + (32)0x00000000]
xor eax, eax
jmp byte (8)0x00000000
leave
ret
0x01000066