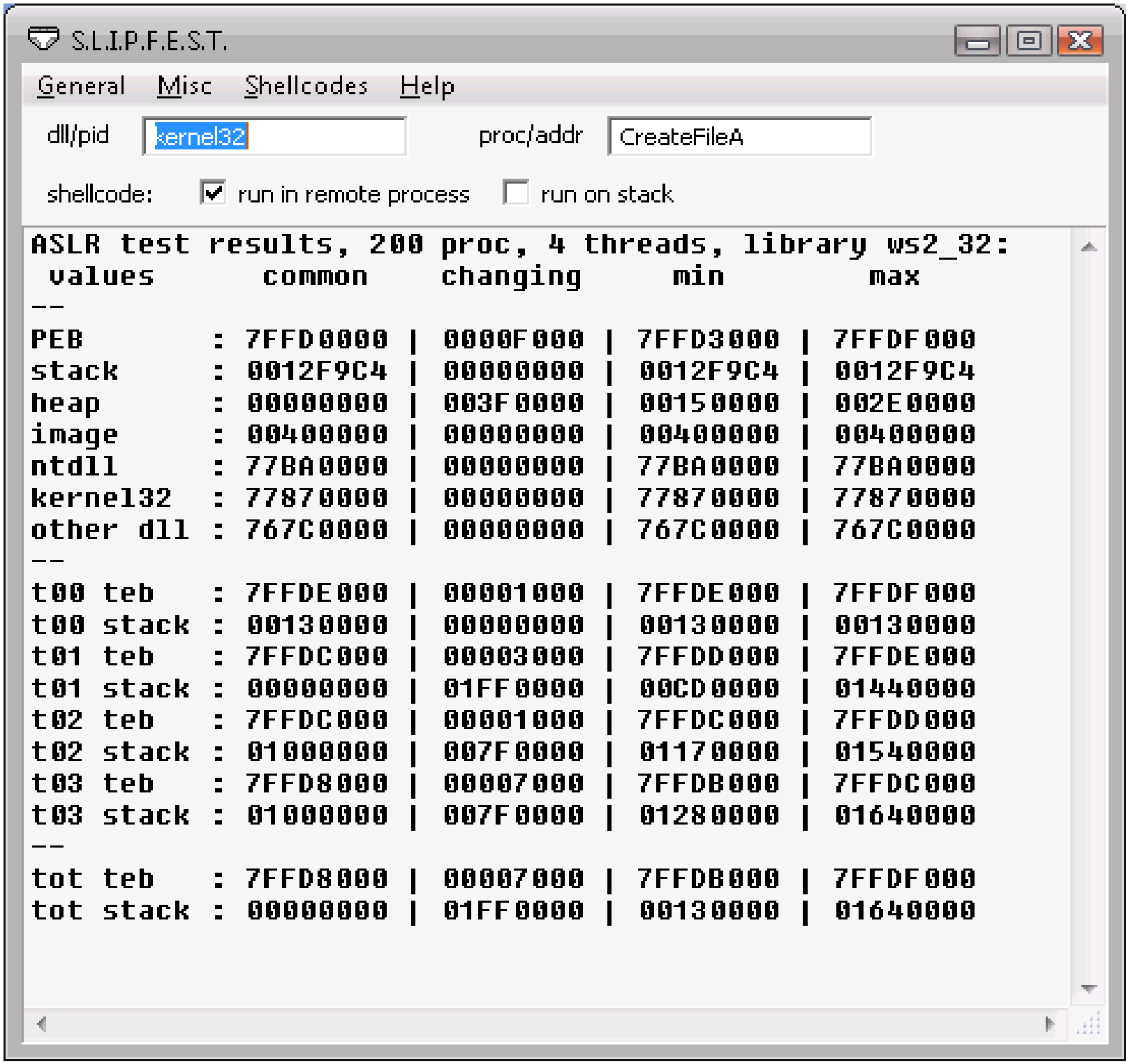
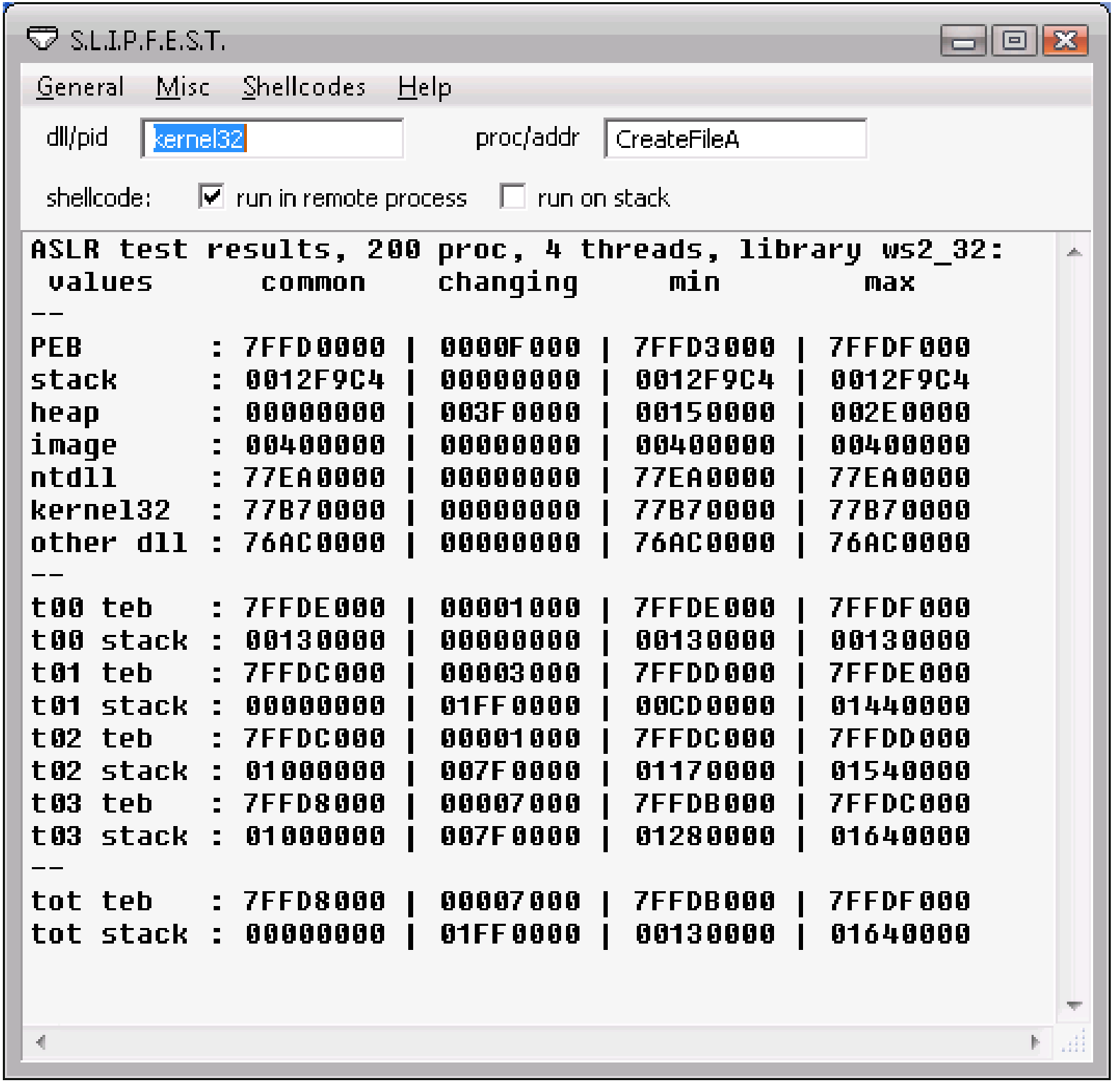
ASLR – Windows Vista
Cas de contournement de protection

ASLR = Address Space Layer randomization

- Fonctionnalité PaX reprise dans Vista
- Principe : 1 application = 1 process principal et souvent plusieurs DLL
- chargement du process, des dlls, de la stack, du heap à des adresses arbitraires
- ASLR = “protection” contre l'exploitation de la plupart des buffers overflows distants (couplé avec NX)


S.L.I.P.F.E.S.T.
General
Misc
Shellcodes
Help
dll/pid
kernel32
proc/addr
CreateFileA
shellcode:
run in remote process
run on stack
ASLR test results, 200 proc, 4 threads, library ws2_32:
values common changing min max
--
PEB : 7FFD0000 | 0000F000 | 7FFD3000 | 7FFDF000
stack : 0012F9C4 | 00000000 | 0012F9C4 | 0012F9C4
heap : 00000000 | 003F0000 | 00150000 | 002E0000
image : 00400000 | 00000000 | 00400000 | 00400000
ntdll : 77BA0000 | 00000000 | 77BA0000 | 77BA0000
kernel32 : 77870000 | 00000000 | 77870000 | 77870000
other dll : 767C0000 | 00000000 | 767C0000 | 767C0000
--
t00 teb : 7FFDE000 | 00001000 | 7FFDE000 | 7FFDF000
t00 stack : 00130000 | 00000000 | 00130000 | 00130000
t01 teb : 7FFDC000 | 00003000 | 7FFDD000 | 7FFDE000
t01 stack : 00000000 | 01FF0000 | 00CD0000 | 01440000
t02 teb : 7FFDC000 | 00001000 | 7FFDC000 | 7FFDD000
t02 stack : 01000000 | 007F0000 | 01170000 | 01540000
t03 teb : 7FFD8000 | 00007000 | 7FFDB000 | 7FFDC000
t03 stack : 01000000 | 007F0000 | 01280000 | 01640000
--
tot teb : 7FFD8000 | 00007000 | 7FFDB000 | 7FFDF000
tot stack : 00000000 | 01FF0000 | 00130000 | 01640000


S.L.I.P.F.E.S.T.
General
Misc
Shellcodes
Help
dll/pid
kernel32
proc/addr
CreateFileA
shellcode:
run in remote process
run on stack
ASLR test results, 200 proc, 4 threads, library ws2_32:
values common changing min max
PEB : 7FFD0000 | 0000F000 | 7FFD3000 | 7FFDF000
stack : 0012F9C4 | 00000000 | 0012F9C4 | 0012F9C4
heap : 00000000 | 003F0000 | 00150000 | 002E0000
image : 00400000 | 00000000 | 00400000 | 00400000
ntdll : 77EA0000 | 00000000 | 77EA0000 | 77EA0000
kernel32 : 77B70000 | 00000000 | 77B70000 | 77B70000
other dll : 76AC0000 | 00000000 | 76AC0000 | 76AC0000
t00 teb : 7FFDE000 | 00001000 | 7FFDE000 | 7FFDF000
t00 stack : 00130000 | 00000000 | 00130000 | 00130000
t01 teb : 7FFDC000 | 00003000 | 7FFDD000 | 7FFDE000
t01 stack : 00000000 | 01FF0000 | 00CD0000 | 01440000
t02 teb : 7FFDC000 | 00001000 | 7FFDC000 | 7FFDD000
t02 stack : 01000000 | 007F0000 | 01170000 | 01540000
t03 teb : 7FFD8000 | 00007000 | 7FFDB000 | 7FFDC000
t03 stack : 01000000 | 007F0000 | 01280000 | 01640000
tot teb : 7FFD8000 | 00007000 | 7FFDB000 | 7FFDF000
tot stack : 00000000 | 01FF0000 | 00130000 | 01640000

Good , but ...

- Problème : Mauvaise randomization (1/256)
- Randomization == reboot (exploits locaux toujours présents)
- 1/256 = possibilité d'exploiter en remote un cgi ou un thread (sans crash du process)