- Per-system randomization provides for random addresses when comparing different systems.
- Per-program randomization provides random addresses for concurrent users of a program.

Per-process randomization provides the best protection against attackers, but is the most costly, both in real memory usage and in the performance of the AIX Virtual Memory Manager (VMM), especially for randomizing shared library addresses.

In summary, with AIX 7.2 TL 3 Service Pack 1, the following enhancements in support of ASLR were implemented:

- The OS can randomize the address-space layout of shared libraries and marked programs.
- Some selected AIX programs are marked so that their address spaces may be randomized if ASLR is enabled by the system administrator.

System administrators may control ASLR as follows:

- Enable randomization of shared library addresses that are used by all programs.
- Enable ASLR for the programs that are marked to allow randomization.
- Allow randomization for new programs or extra existing programs by using the **-baslr** option with the **ld** or **ldedit** command.

The remainder of this chapter provides more information regarding ASLR implementation and control.

2.6.1 Process address space randomized entities in AIX

Beginning with AIX 7.2 TL 3 Service Pack 1, randomization of different parts of the process address space is supported. The following sections cover the randomization techniques that are used in AIX ASLR for the following process address space entities:

- Main program text
- Main program data
- Main stack
- Shared library text
- Shared library data
- Privately loaded libraries

Main program text

The main program text contains position-independent code and can be relocated easily, except for programs that are compiled by the **roptr** (places constant pointers in read-only storage), **ro** (places string literals in read-only storage), and **roconst** (places constants in read-only storage) compiler options. These options allow many address constants to be moved from the data section to the text section, and require that programs be link-edited at their load-time addresses because they have text-section relocations.

A main program is made addressable by mapping the executable file segment into Effective Segment ID (ESID) 0x1 for a 32-bit program and ESID 0x10 for a 64-bit program. Multiple processes running the same program automatically share the main program's text because they use the same Segment ID (SID).