

1. Run with seeds 1, 2, and 3, and compute whether each virtual address generated by the process is in or out of bounds. If in bounds, compute the translation.

ARG seed 1
ARG address space size 1k
ARG phys mem size 16k

Base-and-Bounds register information:

Base : 0x0000363c (decimal 13884)
Limit : 290

The physical limit is 14174, anything large than this is segfault

Virtual Address Trace

VA 0: 0x0000030e (decimal: 782)--> Segfault physical 14666 > 14174
VA 1: 0x00000105 (decimal: 261)--> legal physical 14145 < 14174
VA 2: 0x000001fb (decimal: 507)--> Segfault physical 14391 > 14174
VA 3: 0x000001cc (decimal: 460)--> Segfault physical 14344 > 14174
VA 4: 0x0000029b (decimal: 667)--> Segfault physical 14551 > 14174

ARG seed 2
ARG address space size 1k
ARG phys mem size 16k

Base-and-Bounds register information:

Base : 0x00003ca9 (decimal 15529)
Limit : 500

The physical limit is 16029, anything large than this is segfault

Virtual Address Trace

VA 0: 0x00000039 (decimal: 57)--> legal physical 15586 < 16029
VA 1: 0x00000056 (decimal: 86)--> legal physical 15615 < 16029
VA 2: 0x00000357 (decimal: 855)--> Segfault physical 16384 > 16029
VA 3: 0x000002f1 (decimal: 753)--> Segfault physical 16282 > 16029
VA 4: 0x000002ad (decimal: 685)--> Segfault physical 16214 > 16029

ARG seed 3
ARG address space size 1k
ARG phys mem size 16k

Base-and-Bounds register information:

Base : 0x000022d4 (decimal 8916)
Limit : 316

The physical limit is 9232, anything large than this is segfault

Virtual Address Trace

VA 0: 0x0000017a (decimal: 378)--> Segfault physical 9294 > 9232
VA 1: 0x0000026a (decimal: 618)--> Segfault physical 9534 > 9232
VA 2: 0x00000280 (decimal: 640)--> Segfault physical 9556 > 9232
VA 3: 0x00000043 (decimal: 67)--> legal physical 8983 < 9232
VA 4: 0x0000000d (decimal: 13)--> legal physical 8929 < 9232

2. Run with these flags: `-s 0 -n 10`. What value do you have set `-l` (the bounds register) to in order to ensure that all the generated virtual addresses are within bounds?

ARG seed 0
ARG address space size 1k
ARG phys mem size 16k

Base-and-Bounds register information:

Base : 0x00003082 (decimal 12418)
Limit : 472

Virtual Address Trace

VA 0: 0x000001ae (decimal: 430) --> PA or segmentation violation?
VA 1: 0x00000109 (decimal: 265) --> PA or segmentation violation?
VA 2: 0x0000020b (decimal: 523) --> PA or segmentation violation?
VA 3: 0x0000019e (decimal: 414) --> PA or segmentation violation?
VA 4: 0x00000322 (decimal: 802) --> PA or segmentation violation?
VA 5: 0x00000136 (decimal: 310) --> PA or segmentation violation?
VA 6: 0x000001e8 (decimal: 488) --> PA or segmentation violation?
VA 7: 0x00000255 (decimal: 597) --> PA or segmentation violation?
VA 8: 0x000003a1 (decimal: 929) --> PA or segmentation violation?
VA 9: 0x00000204 (decimal: 516) --> PA or segmentation violation?

`python3 relocation.py -s 0 -n 10 -l 930`

The largest one plus one which in this case is decimal $929 + 1 = 930$. Then we can ensure that all the generated virtual addresses are within bounds.

3. Run with these flags: `-s 1 -n 10 -l 100`. What is the maximum value that base can be set to, such that the address space still fits into physical memory in its entirety?

ARG seed 1
ARG address space size 1k
ARG phys mem size 16k

Base-and-Bounds register information:

Base : 0x00000899 (decimal 2201)
Limit : 100

Virtual Address Trace

VA 0: 0x00000363 (decimal: 867) --> PA or segmentation violation?
VA 1: 0x0000030e (decimal: 782) --> PA or segmentation violation?
VA 2: 0x00000105 (decimal: 261) --> PA or segmentation violation?

VA 3: 0x000001fb (decimal: 507) --> PA or segmentation violation?

VA 4: 0x000001cc (decimal: 460) --> PA or segmentation violation?

For this problem, the maximum space is of physical memory is 16k which is equal to $16 * 1024$ (1k = 1024 bytes) minus the limit (we need save the space for the process that does not go out physical memory range) thus we got:

$16 * 1024 - 100 = 16284$ Then our command as follow:

python3 relocation.py -s 1 -n 10 -l 100 -b 16284 -c

4. Run some of the same problems above, but with larger address spaces (-a) and physical memories (-p).

python3 relocation.py -s 1 -n 10 -l 100 -a 32m -p 64m -c -b 67108764

this time we calculate maximum base space before 100 limit is

$64 * 1024 * 1024 - 100 = 67108764$

- 4. What fraction of randomly-generated virtual addresses are valid, as a function of the value of the bounds register? Make a graph from running with different random seeds, with limit values ranging from 0 up to the maximum size of the address space.**

Since every randomly generated virtual address with in the virtual address size, and once the bounds register grows close to the randomly-generated virtual address, the probability for virtual process within the limit is increase, and once reach to virtual address size, the fraction become 1 which mean 100% in limit.

