Project 1.5: Detecting Heap Buffer Overflow Using Canary

Concept

Similar to stack canary, we also save a random canary around the heap buffer (either at the head or end of the buffer, depending on the robustness and security) upon heap allocation (i.e., malloc()). When a heap buffer overflow occurs, the random canary will be overwritten by other values. We trigger the detection when the buffer is freed -- obtain the value at the canary location and compare it with the original canary. If they are not equal, we report a heap buffer overflow.

Steps

1. Check the files: overflow.c contains a heap overflow vulnerability. The file detector.c is the one you need to work on, which contains three TODOs for you. The Makefile is to make your life easier.
2. General use:
   1. Use “$ **make all**” to build your code
   2. Use “$ **make clean**” to clean your build
   3. Use “$ **make test**” to run and test your detector
   4. Read the commands in the Makefile for more details.
3. Finish the “TODOs” in detector.c. If your detector works, you should be able to catch the overflow by running “$ make test”.

Important Notes

* Do not use cryptographically insecure random number generators to get the canary.
  + That is, do **not** use srand() or rand()
  + Use /dev/random or /dev/urandom
* Your detector should be able to detect the overflow even only one byte is overwritten.
* We assume the overflow direction is from low address to high address.
* You may reuse the VM given for Project 1
* You may install useful tools like Eclipse, SSH, or Vim, but make sure that the Makefile works outside of these tools. We will be compiling the Makefile using only the command line on Ubuntu.
* You are not allowed to use dependencies that do not already exist naturally on the Project 1 VM image we gave you.

References

* Some information about canaries: <https://en.wikipedia.org/wiki/Buffer_overflow_protection>
* Chapter 10 of the 3rd edition of the Stallings-Brown textbook.
* <https://www.usenix.org/legacy/publications/library/proceedings/sec98/full_papers/cowan/cowan.pdf>

Deliverables

* detector.c