Alex Hoff

Networks Lab 5 Write Up

Submit well commented generator and entropy code, and be sure to cite any sources that you use (especially if you make use of the sample code from Cisco). Include measurements of entropy for different file sizes using your generator program. But also include the results of measuring the entropy of at least three sample files of your choosing. For example, testing an image file that is in a compressed format would be expected to produce a higher entropy value than a file that is all zeros. As part of your README, be sure to discuss your results for these three files in addition to how you chose to generate the random byte stream.

Entropy Tests from Rand Byte Generator

1 K

Entropy is 7.811686

10 K

Entropy is 7.984531

1 MB

Entropy is 7.999838

5 MB

Entropy is 7.999968

These values were just as we expected. Larger files had many more bytes so a higher chance of having a wider range of bytes. Therefore the large files had entropy almost exactly to 8, the number of bits in a byte while the smaller files did not.

Entropy Tests for 5 K sample files

.jpg

Entropy is 7.919843

.tar

Entropy is 1.238018

.cpp

Entropy is 4.857694

These values were a little less expected. The jpg file was as expected with a very high entropy as image files are compressed files that take a stream of bits and try to take out all redundancies therefore there would be a large range of unique values. The cpp file was written byNetworks Lab5 me and had a medium entropy because it used a lot of similar variables. The tar file stumped me for a second why it was so low but upon more research we found that tar files have headers and blocks at the end which are stuffed with zeros. Additionally tar implementations fill extra space with zeros so there were many more zeroes than any other number in the file lowering the entropy drastically. ([https://en.wikipedia.org/wiki/Tar\_%28computing%29](https://en.wikipedia.org/wiki/Tar_(computing)))

I implemented by random byte generator based off what I could find on the internet. I knew I could use rand() to create random numbers as I have done before so I made rand moduloed with 256 which is 2 to the 8th, for 8 bits in a byte, and then outputted them with a variable so I could change the size of the output.

Sources:

blogs.cisco.com/security/on\_information\_entropy

* for creating the entropy file
* used the algorithms given to calculate entropy
* used the file buffer loop

<http://stackoverflow.com/questions/15621764/generate-random-byte-stream-with-c>

* used to help create the rand byte generator

How to run:

gcc randgen.c -o randgen

gcc entropycalc.c -lm -o entropycalc

./randgen | entropycalc