Software Bertillonage: Finding the Provenance of an Entity

Summary

This paper discussed a way of identifying the origin of a program based on an inspiration from an old way of identifying suspects. It uses the idea that software with similar object of interests (OOI) would highly likely be similar. They used tools to extract information base on OOI for source code and decompiled source code. Then tested the results on the *java Maven* repository. The result showed that the method could achieve very high rate (~96%) of identifying the provenance for decompiled source code of 2 binary files. The result for decompiled source code and original source code is not as accurate as source code but is still quite impressive (~68%). One main reason for the lower accuracy maybe related to the way binary is decompiled.

Things I would like to see discussed

- The results seems to imply that between versions of software, many OOIs are modified. So what is a version change in general. Is it a convention or they are just per person.
- What causes the difference between decompiled and source code? I'm not very familiar on this topic.
- Malware detection potential? Since binary to binary seems to have a great precision it would detect differences between original file.