

CS380S: Project Proposal

Rushi Shah

Andrew Russell

1 Project Idea

We plan to use Data Dependency tools to determine how system entropy in a given program is used by various cryptographic algorithms. That will allow us to identify if and when entropy is too low or is misused. We can release this as a tool for developers to use as part of a compiler toolchain. We can also use the tool on current projects that might be misusing crypto/entropy.

We have two motivating examples. First, we would like our tool to be able to detect the issue with the Debian/OpenSSL pseudo-random number generator that was exposed in 2008. Second, we would like to identify potential vulnerabilities in current cryptocurrency wallet code.

2 Rough Plan

1. Determine data dependency tool (for C/C++) to use, conduct background research (*2 weeks*)
2. Adapt tool to identify the OpenSSL bug (*6 weeks*)
3. Test on other cases like bitcoin wallets (*2 weeks*)
4. Prepare presentation/writeup (*1 week*)
5. Integrate tool into compiler toolchain like clang (*optional*)

3 Research Hypothesis

1. An automated tool can detect entropy bugs in real-world programs.
2. Entropy is insufficiently propagated in programs that rely on cryptography.

4 Related Work

4.1 Background information

1. Debian/OpenSSL Bug
 - (a) https://www.schneier.com/blog/archives/2008/05/random_number_b.html
 - (b) <https://research.swtch.com/openssl>
 - (c) <https://freedom-to-tinker.com/2013/09/20/software-transparency-debian-openssl-bug/>
 - (d) <https://www.cs.umd.edu/class/fall2017/cmsc8180/papers/private-keys-public.pdf>
2. Data flow
 - (a) https://en.wikipedia.org/wiki/Data-flow_analysis
 - (b) <https://www.seas.harvard.edu/courses/cs252/2011sp/slides/Lec02-Dataflow.pdf>
3. Static Program Analysis
 - (a) <https://cs.au.dk/~amoeller/spa/spa.pdf>

4.2 Related research