Analysis of randomness levels in the kernel entropy pool after boot

Thesis Abstract

Peter Kotvan

Pseudo-random generators (PRNG) are crucial component of modern cryptosystems. There are several applications in which PRNGs are required e.g. cryptographic key generation, cryptographic nonces, salt or one-time pads. With requirements of higher security of cryptosystems the necessity of high quality random data is apparent. Since computers are deterministic machines it is very hard to produce actually random numbers on them. With pseudo-random generators it should be possible to generate numbers with statistical properties of random data using an algorithm. In practice PRNG uses a random seed, that is used to initialize the PRNG, however knowledge of the seed allows the attacker to derive the data generated by the PRNG.

To ensure, that the seed is unpredictable by the attacker it can be gathered from different sources of the "environment noise". These sources produce entropy that is collected and used as an input to the PRNG, for example:

- the system clock;
- elapsed time between keystrokes or mouse movement;
- content of input and output buffers;
- user input; and
- operating system values such as system load and network statistics.

In the UNIX world it is the kernel that has direct access to the hardware and hence the entropy collection is best done from inside the kernel.

In this thesis I will analyse the quality of data used as the input of Linux kernels internal PRNG. I chose this operating system because it is often used in situations where the reliable source of random numbers play crucial role in the security. It mostly includes services provided by servers but also desktops computers, routers and others. The entropy produced by the system varies depending on the specific hardware used. In my research I've focused on systems with restricted sources of entropy specifically virtual machines but our findings may apply to other devices without input/output devices and with few hardware components providing interrupts.

Since the Linux kernel is released under open source license I was able to modify the source code to be able to collect the data from entropy sources. I have executed two data collections, one on six identical virtual machines and the other in an openstack instance on 12 virtual machines.

Openstack is a free and open-source cloud computing software platform, that provides infrastructure as a service (IaaS). In this environment it is simple to spawn new virtual machines on demand. I have created minimal image of Fedora Linux distribution version 21 containing the kernel with my modifications and used with openstack. I've programmed simple monitoring script that monitored the virtual machines and collected the data samples.

The analysis of the data showed that data from some sources are redundant (they are always the same) and hence do not contribute any entropy to the PRNG. On the other hand hardware interrupts are sources of data suggesting hi level of randomness. In following work I will do further analysis of the data. According to the NIST document 800-90B Recommendation for the Entropy Sources Used for Random Bit Generation I want perform Min-Entropy estimation. I am also considering the use of NIST Statistical Test Suite. There is an open source optimised version of this test suite available claiming to be 30 times faster as the one provided by the NIST (https://github.com/sysox/NIST-STS-optimised).

The goal of the thesis is to show whether the entropy gathered from the environment is sufficiently random and unpredictable even for devices with constrained hardware.