Kernel CPU Time Jitter RNG Entropy Source version 3.4.0 kernel space

SP 800-90B Non-Proprietary Public Use Document

Document Version: 1.0

Document Date: 2023-08-20

Prepared by:
atsec information security corporation
9130 Jollyville Road, Suite 260
Austin, TX 78759
www.atsec.com

Table of Contents

1 Description

The Kernel CPU Time Jitter RNG version[1] “3.4.0 kernel space” is a non-physical entropy source that is implemented in the kernel binary. Its purpose is to feed the kernel DRBGs. The noise generation of this entropy source is based on the tiny variations in the execution time of the same piece of code. The execution time of this piece of code is made unpredictable by the complexity of the different hardware components that comprise modern CPUs and the different internal states that the operating system can have at a certain point in time.

The entropy source was tested on the operational environments listed in Table 1. The noise source was tested under the assumption that its output is non-IID.

Table 1: Operational environment and version.

Manufacturer	Model	Operational Environment and Version	Processor
Amazon Web Services (AWS)	m5.metal	AlmaLinux 9.2	Intel Xeon Platinum 8259CL
Amazon Web Services (AWS)	a1.metal	AlmaLinux 9.2	AWS Graviton

2 Security Boundary

The boundary for this non-physical, software-based entropy source is the executable binary file. It is compiled from the C code that implements it (i.e., kernel source code).

Figure 1 depicts the overall design of the entropy source and its core operations.

[1] The source is composed of CPU Jitter version 2.2.0 in kernel space and the following patches.
https://github.com/torvalds/linux/commit/bb897c55042e9330bcf88b4b13cbdd6f9fabdd5e
https://github.com/torvalds/linux/commit/3fde2fe99aa6dacd4151c87382b07ce7f30f0a52

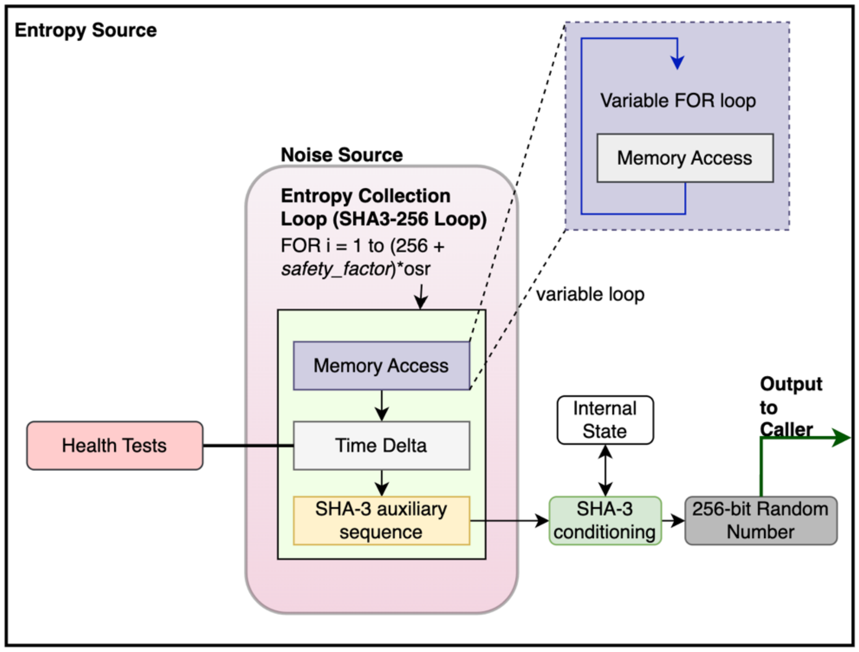

Figure 1: CPU Jitter 3.4.0 kernel space Design

The noise source is implemented by collecting and accumulating time variances of variable memory accesses and variances in the execution time of a defined set of instructions, which includes an implementation of SHA3-256. The time variances, in the form of time deltas, are accumulated and mapped down to 256-bits by the SHA3-256 vetted conditioning function which outputs 256 bits of full entropy.

If the Repetition Count Test (RCT) or the Adaptive Proportion Test (APT) health tests fail, the noise data is discarded, the entropy source halts without outputting any data, and a failure code is returned to the caller. If the health test failure is permanent, the kernel which contains this entropy source will panic.

3 Operating Conditions

The noise source is non-physical, and thus the operating conditions are inherited from the operational environment in which the entropy source is installed, as shown in Table 2 below.

Table 2: Operating Conditions for each Operational Environment

Manufacturer / Model	Temperature	Voltage	Humidity	Clock Speed	Cache Sizes
AWS m5.metal	10C° – 35C°	+100-240V	5 to 95%	2.5 GHz	L1d: 48x32 KB L1i: 48x32 KB L2: 48x1 MB L3: 2x35.75 MB
AWS a1.metal	10C° – 35C°	+100-240V	10 to 90%	2.3 GHz	L1d: 16x32 KB L1i: 16x48 KB L2: 4x2 MB

4 Configuration Settings

No configuration settings are required.

5 Physical Security Mechanisms

The noise source is non-physical. The physical security mechanisms only apply to the hardware component of the operational environment in which the entropy source is installed, and thus the entropy source inherits those mechanisms.

6 Conceptual Interfaces

The entropy source provides the following interfaces:

- jent_kcapi_random(): Obtains conditioned entropy for the caller. This is the main function of the entropy source, the one that shall be used to request entropy data. This interface corresponds to the GetEntropy() conceptual interface from SP800-90B.
- jent_measure_jitter(): Obtains raw noise data for testing purposes. This interface corresponds to the GetNoise() conceptual interface from SP800-90B.

7 Min-Entropy Rate

The noise source provides an entropy rate of $H_{submitter}$ = 1 bit per 64-bit time delta.

The entropy source collects 320 time deltas of 64 bits each (20480 bits) as input to the SHA3-256 conditioning component. This corresponds to 320 bits of entropy. The output entropy rate of the SHA3-256 is assessed to be 1 bit/bit.

As a result, the entropy source provides 1 bit/bit of entropy rate at its output.

8 Health Tests

The entropy source implements the following continuous health tests:

- Repetition Count Test conforming to SP 800-90B section 4.4.1.
  - $H = 1$ bit of entropy per 64-bit time delta
  - Intermittent failure alpha value $\alpha_i = 2^{-30}$
  - Permanent failure alpha value $\alpha_p = 2^{-60}$
  - Intermittent failure cutoff value $C_i = 31$
  - Permanent failure cutoff value $C_p = 61$
- Adaptive Proportion test conforming to SP 800-90B section 4.4.2.
  - $W = 512$
  - $H = 1$ bit of entropy per 64-bit time delta
  - Intermittent failure alpha value $\alpha_i = 2^{-30}$
  - Permanent failure alpha value $\alpha_p = 2^{-60}$
  - Intermittent failure cutoff value $C_i = 325$
  - Permanent failure cutoff value $C_p = 355$

- Stuck (Non-Permanent) Test: The stuck test computes the first, second and third discrete derivatives of the time value that will be processed by the SHA3-256. If any of these derivatives are zero, then the received time delta is considered stuck. In this case the input state to SHA3-256 is not updated, and the entropy value is not counted. The stuck test then triggers the RCT for further processing. The second derivative is in fact the RCT itself.

As allowed by Section 4.3 of SP 800-90B, the entropy source defines two types of health test failures for the RCT and the APT: intermittent failures and permanent failures.

An intermittent failure is characterized by a false positive probability $\alpha_i = 2^{-30}$, which lies within the recommended range of $2^{-20} \leq \alpha \leq 2^{-40}$. When an intermittent failure is detected, the CPU Jitter RNG is automatically reset (which includes clearing the entropy pool and resetting the DRBG conditioning component), and the caller is notified of this failure. The only exception to this rule is during the start-up tests, where intermittent failures will be treated as permanent.

Permanent failures are characterized by a false positive probability of $\alpha_p = 2^{-60}$. When a permanent failure is detected, the CPU Jitter RNG is also reset, but the Linux kernel that contains this entropy source immediately enters the error state. In practice, this results in a kernel panic.

The continuous-health tests are applied to each new time delta obtained from the noise source.

The stuck test is considered non-permanent, as positive stuck tests will be registered but will not immediately halt the entropy source.

Start-up tests conduct the same set and parameters of the continuous health tests on 1024 time deltas. The data is discarded after the start-up tests have completed successfully. Any health test failure during the start-up tests will always be treated as a permanent failure, which results in the permanent shutdown of the entropy source.

On-demand health tests of the noise source may be performed by rebooting the operational environment, which results in the immediate execution of the start-up tests. Similar to the start-up tests, the data used for the on-demand health tests are discarded after successful completion.

When entropy is requested from through the jent_kcapi_random() function, the return code indicates the status of the entropy source. A return code of -EAGAIN indicates an intermittent health test failure, whereas a return code of -EFAULT indicates a permanent health test failure.

9 Maintenance

There are no maintenance requirements as this is a software-based entropy source.

10 Required Testing

To test the entropy source, raw data samples must be collected using a test harness that is capable of accessing the jent_measure_jitter() noise interface from the entropy source. The test harness and accessory kernel tools must be supplied by the vendor.

Raw noise data consisting of at least 1,000,000 64-bit samples truncated to 8 bits must be collected from the operational environment at its normal operating conditions and processed by the SP 800-90B entropy tool that is provided by NIST. The expected min-entropy rate must approach the one in Section 7.

Restart data must be collected at normal operating conditions through the jent_measure_jitter() interface following the restart procedure specified in SP 800-90B (i.e., 1,000 samples from 1,000 restarts each) and processed by the NIST SP 800-90B entropy tool. The minimum of the row-wise and column-wise entropy rate must be more than half that of the raw noise entropy rate.