Forcepoint NGFW Entropy Library

SP 800-90B NON-PROPRIETARY PUBLIC USE DOCUMENT

DOCUMENT VERSION 0.3

ENTROPY SOURCE VERSION 3.4.1

10900-A Stonelake Blvd.
Austin, TX 78759, USA
www.forcepoint.com

Revision History

Revision	Date	Reason
0.1	July 05, 2023	Initial release
0.2	October 31, 2023	Table updates and clarifications to Required Testing and terminology
0.3	February 21, 2024	Addressed review comments

Trademarks, Copyrights, and Third-Party Software

Contents

Description

The Forcepoint NGFW Entropy Library version 3.4.1 is a non-physical, non-IID entropy source based on the open-source jitterentropy-library ("Jitter RNG") software library version 3.4.1. The implementation collects entropy from jitter (deltas of high-resolution timestamp values) in the timing of memory accesses and mixes the entropy into a 256-bit entropy pool using SHA3-256 for the mixing function.

Table 11 lists the supported operating environments with which the entropy source implementation has been tested.

Model	Hardware	Revision	CPU	Operating System
N60	60-C1	1	Intel Atom® C3338R (Goldmont)	NGFW OS 10 on Linux 4.19
N120	120-C3	1	Intel Atom® C3338R (Goldmont)	NGFW OS 10 on Linux 4.19
N120L	120-C4	1	Intel Atom® C3338R (Goldmont)	NGFW OS 10 on Linux 4.19
N352	352-C1	1	Intel Atom® C5315 (Tremont)	NGFW OS 10 on Linux 4.19
N355	355-C1	1	Intel Atom® C5325 (Tremont)	NGFW OS 10 on Linux 4.19

TABLE 11 TESTED OPERATING ENVIRONMENTS

Security Boundary

The whole entropy source implementation lies inside a security boundary. The Forcepoint NGFW Entropy Library implementation consists of a binary compiled from source code and the public API methods with which the entropy source can be initialized, controlled, and the output entropy can be obtained by the caller.

Table 11 shows the design of the entropy source and the entropy-relevant operations. The entropy collection loop is run *(256 + safety factor) * oversampling rate* times, where *safety factor (sf)* is 64 in the Approved Mode and *oversampling rate (osr)* is 3 for each iteration to provide at least *1/osr* bits of entropy into the pool.

The output of the entropy source is used to seed SP 800-90A compliant cryptographic DRBGs which are located outside the security boundary.

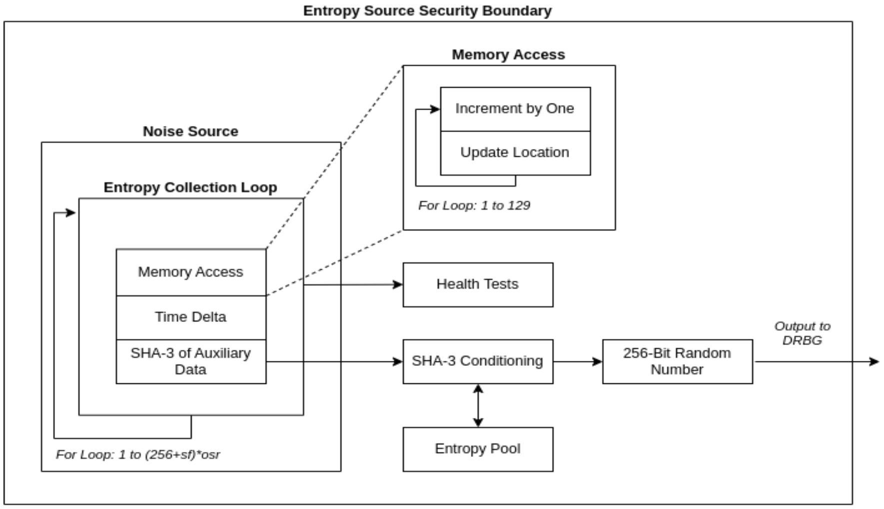

FIGURE 11 ENTROPY COLLECTION OPERATION

Operating Conditions

Table 22 lists the operating conditions for the tested hardware platforms.

Model	Operating Temperature	Operating Voltage	Clock Speed
N60	0° to 40° C	12 VDC	1800 MHz
N120	0° to 40° C	12 VDC	1800 MHz
N120L	5° to 40° C	12 VDC	1800 MHz
N352	0° to 40° C	19 VDC	2400 MHz
N355	0° to 40° C	19 VDC	2400 MHz

TABLE 22 OPERATING CONDITIONS

Configuration Settings

The Forcepoint NGFW Entropy Library implementation uses the Jitter RNG software library version 3.4.1 which has been compiled from unmodified source code.

Table 33 lists the entropy-relevant parameters which must be preserved to comply with the ESV certificate.

Parameter Type	Parameter Name and Value	Description
Constant definition	ENTROPY_SAFETY_FACTOR = 64	The applied additional safety factor (*sf*) when Approved Mode is enabled
	JENT_MIN_OSR = 3	The applied oversampling rate (*osr*)
	JENT_CONF_DISABLE_LOOP_SHUFFLE = True	Disable the random number of hash loops performed for each call, loop count is fixed to 1
	JENT_CONF_ENABLE_INTERNAL_TIMER = True	Enable internal timer support

Kernel command line parameter	fips = 1	Approved Mode is enabled

TABLE 33 ENTROPY-RELEVANT PARAMETERS

Physical Security Mechanisms

As Forcepoint NGFW Entropy Library is based on non-physical software implementation, all physical security mechanisms are inherited from the hardware platform on which the entropy source is operating and are applied only to the hardware platform.

Conceptual Interfaces

Table 44 describes the conceptual interfaces provided by Forcepoint NGFW Entropy Library.

Interface	Description
GetEntropy	The API method jent_read_entropy() is used to obtain the requested amount of conditioned entropy, corresponding to the GetEntropy interface as described in SP 800-90B section 2.3.1
GetNoise	No specific GetNoise API method exists but an auxiliary test harness can be used to collect raw noise data using the API method jent_hash_time() as allowed in SP 800-90B section 2.3.1. The test harness collects 1 million samples of 64-bit data and 1000 samples of 64-bit data with 1000 restarts.
HealthTest	No specific HealthTest API method exists but the health tests can be executed by allocating a new entropy collection handle with the method jent_entropy_collector_alloc() as allowed in SP 800-90B section 2.3.1

TABLE 44 CONCEPTUAL INTERFACES

Min-Entropy Rate

The entropy rate of the included noise source for each 64-bit input time delta value is $H_{submitter} = 1/3 \approx 0.333$ *bits*.

Due to the use of the safety factor in the Approved Mode and SHA-3 256 as the vetted conditioning component the Forcepoint NGFW Entropy Library implementation produces output that is considered to contain full entropy, that is the 256-bit output of a call to the GetEntropy conceptual interface contains $H_{out} = 256$ *bits* of entropy.

Health Tests

As required in SP 800-90B section 4.3 Forcepoint NGFW Entropy Library performs health tests when the entropy source instance is initialized and performs continuous health tests during operation. When any health test fails the entropy data block causing the failure is discarded before being returned to the caller and the caller is informed of the failure by setting the appropriate error code depending on the failing test.

Any health test failure will also render the entropy source instance permanently unusable due to entering an error state and a new entropy source instance must be initialized to obtain further entropy output.

The following health tests are available:

- Repetitive Count Test (RCT), as described in SP 800-90B section 4.4.
  - Entropy per 64-bit sample: $H=0.333$ *bits*
  - Alpha value: $\alpha=2^{-30}$
  - Cutoff value: $C=91$
- Adaptive Proportion Test (APT), as described in SP800-90B section 4.4.
  - Entropy per 64-bit sample: $H=0.333$ *bits*
  - Alpha value: $\alpha=2^{-30}$

  - Cutoff value: *C=459*
  - Window size: *W=512*
- Stuck test, a vendor designed test which calculates the first, second and third discrete derivative of the time delta to be processed by the hash.
  - Only if all derivative values are non-zero is the time delta considered to be non-stuck, otherwise the time delta is rejected.
- Lag Predictor Test, a vendor-designed test based on SP 800-90B section 6.3.8.
  - Window size: *W=131072*
  - History size: *lag_history_size=8*
  - Alpha value: $\alpha=2^{-30}$

Conditioning Component

Forcepoint NGFW Cryptographic Kernel Module uses the SHA3-256 hashing function as the conditioning component. The SHA3-256 (Cert #A4426) is considered a vetted conditioning function as described in SP 800-90B 3.1.5.1. No cryptographic key is required by the conditioning component.

The conditioning component produces full entropy output (h_{out}): Each 256-bit output block contains 256 bits of entropy. Output blocks from the conditioning component may be concatenated and/or truncated to obtain the requested amount of output entropy.

Maintenance

There are no maintenance requirements for Forcepoint NGFW Entropy Library.

Required Testing

A purpose-built test harness is used to collect raw data for testing. The test harness collects 1 million consecutive raw 64-bit samples and then it collects 1000 raw 64-bit samples over 1000 restarts of the entropy source collection, totaling another 1 million samples.

A checklist for the module to ensure the entropy source is running properly:

1. Raw noise data obtained with the test harness should be processed by the SP800-90B tool to obtain an entropy rate which must be near equal to or the defined min-entropy rate.
2. Restart noise data obtained with the test script should be processed by the SP800-90B tool.
   a. The sanity test to apply to the noise restart data must pass, and
   b. The minimum of the row-wise and column-wise entropy rate shall not be less than half of the entropy rate from 1 above.

The test harness must be provided by the vendor.