René Peralta May 28, 2019
Computer Scientist
National Institute of Standards & Technology (NIST)

Dear Mr. Peralta,

Congratulations to you and your co-authors on draft NISTIR 8213. It is an important industry milestone.

Taproot Security has followed NIST's work on Randomness Beacons for some time and we support its objectives. In fact, I blogged about it earlier this year.[1]

The following comments pertain to the draft version of *A Reference for Randomness Beacons: Format and Protocol Version 2*.[2]

Pulse Format

1. The requirement that the `version` field remain constant in a chain [line 1058] seems to imply that upgrading the Beacon App to a new pulse version requires ending chain(s) using the previous version and starting new chain(s). Alternatively, it may imply the Beacon App must support multiple chains using different pulse versions. The protocol should discuss this.
2. Limiting the `version` field to *x.y* while excluding the minor subversion (*x.y.z*) makes it challenging for a relying party to determine *z*. While we understand *z* is not needed to parse the pulse format, there are other legitimate reasons why a consumer may wish to know its value.

Beacon Operation

3. The draft protocol does not require a Beacon to advertise RNG metadata to consumers, although a Beacon Operator may choose to include such information in Local History [6.6 5l]. We believe it will be critical for many consumers to know entropy sources (e.g., quantum vs. classical) and whether RNGs have passed standard statistical randomness tests.[3] The protocol should require RNG metadata be published by all Beacons.
4. The draft protocol does not require a Beacon to describe its time sources to consumers, although a Beacon Operator may choose to include such information in Local History [6.6 5l]. We believe it will be critical for some consumers to know time source type (e.g., NTP or GPS) and what external source(s) were used (e.g., US Naval Observatory). The draft protocol discusses time sources [8.2.3] but does not address this issue specifically.
5. The draft protocol does not require a Beacon to describe its HSM to consumers, although a Beacon Operator may choose to include such information in Local History [6.6 5l]. We believe it will be important to some consumers to know whether a HSM is in use -- the draft protocol states the HSM is optional [line 2133] -- and its FIPS 140 certification status/level.

[1] https://www.taprootsecurity.com/single-post/2019/01/07/Random-Security
[2] https://csrc.nist.gov/publications/detail/nistir/8213/draft
[3] https://csrc.nist.gov/csrc/media/publications/conference-paper/1999/10/21/proceedings-of-the-22nd-nissc-1999/documents/papers/p24.pdf

6. The draft protocol implies a Beacon may publish two or more chains concurrently. However, it does not state whether the same `randLocal` value can be used in two pulses on separate chains. In our opinion, this should be explicitly disallowed.
7. If a Beacon uses an external randomness source whose period is longer than the local period, it is unclear in the draft protocol whether the Beacon may reuse a previous `randOut` from the external source in local pulses generated during the lull until the next external pulse.

Failure Modes

8. If a Beacon Operator discovers it was compromised in a way that affects trustworthiness, randomness, or security of past `randOut` values, then the protocol should provide a revocation mechanism to terminate affected chain(s), retroactively invalidate affected pulses, and make the action known to relying parties. Something analogous to a certificate revocation list (CRL) may be appropriate.
9. When one RNG becomes temporarily unavailable, is the Beacon permitted to continue publishing pulses if two or more RNGs remain operational? What if only one working RNG remains? In both cases, we recommend a pulse gap until the affected RNG returns to normal operation.
10. The draft protocol says a Beacon SHOULD publish a certificate revocation policy [line 2126]. For a CA-issued signing certificate, we feel the CA MUST be required to publish CRL and/or OCSP. For a self-signed certificate, the Beacon itself MUST be required to do this. In either case, certificates SHOULD include CRL distribution point[4] or OCSP access information[5] X.509 extensions. They SHOULD also set the *digitalSignature* bit in the Key Usage X.509 extension.[6]
11. If its signing certificate (or the issuing CA's certificate) is revoked, is a Beacon expected to stop publishing fresh pulses until the compromised certificate is replaced? Presumably pulses signed with a revoked certificate are not valid. The draft protocol doesn't appear to address this.
12. The draft protocol implies a Beacon may continue emitting pulses after the signing certificate has expired [line 1229]. If this is correct, it contradicts the recommendation that signing certificates expire after no more than five years [line 2124]. Certificate lifetime is meaningless if expiration is unenforced. The protocol should explicitly disallow this.
13. Consumers requiring high availability (HA) may use multiple Beacons to allow their systems to function when any one Beacon is temporarily unavailable. Alternatively, HA applications might only depend on Beacon(s) for the initial seed value of a local PRNG if that provides adequate randomness. The protocol should make recommendations for HA applications.
14. If a consumer combines pulses from multiple Beacons [7.4] the protocol makes no recommendation how to proceed if one of the Beacons stops producing fresh pulses or stops responding altogether. In a 2-Beacon scenario, this may render the consumer temporarily dependent on a single Beacon, and therefore vulnerable to Malicious Beacon attacks. The protocol should recommend 3 or more Beacons, and explain the risks of single Beacon mode.
15. A malicious Beacon App or RNG deliberately manipulating `randLocal` values is a serious concern, particularly for security applications operating in zero trust environments. The XOR solution described in 8.3.1 [line 2249] would considerably mitigate the risk, and therefore increase relying parties' general trust in Beacons. It's unclear to us why this was not included in protocol 2.0.

[4] IETF RFC 5280 4.2.1.13

[5] IETF RFC 5280 4.2.2.1

[6] IETF RFC 5280 4.2.1.3

Beacon Usage

16. As a general comment, we feel the draft document does not adequately describe proper and improper Beacon usage (use cases and misuse cases).
17. The NIST project web page[7] rightly includes a warning "Do not use Beacon generated values as secret cryptographic keys". This warning should also appear in the protocol specification, and should be extended beyond keys to all security secrets (e.g., PINs) since Beacon pulses are public.
18. Since a major driver for randomness is the need to generate keys, passwords, and other security secrets, the protocol should discuss what role (if any) Beacons may play in those use cases. For example, a relying party may choose to use a Beacon pulse to seed a local PRNG and generate pseudorandom cryptographic keys locally. This must be explicitly discouraged since a PRNG's output is generally predictable if the seed value is known.
19. The draft protocol describes using a Beacon to seed a PRNG [lines 1876, 1952, etc.] and even offers a PRNG algorithm for that purpose [Algorithm 7]. As mentioned in our previous comment, it is important that this approach not be used to generate security secrets.
20. As a future enhancement (Additional Consideration) perhaps Beacons could support security secrets if a relying party could request a *private chain*, with queries against that chain only allowed from that authorized consumer, authenticated via mutual TLS, VPN, IPsec, etc. Since a breach of the Beacon Database could expose that private chain, it could be encrypted using a key known only by the relying party. The relying party may also want the capability to purge a pulse from the database after retrieval.

Minor Comments

21. As depicted in Figure 1, the External Entropy source should connect to the RNG, not to the Beacon App. Or, if this was intentional, please explain in the text.
22. The *cid* referenced in Table 10 row 3h is called a "certID" in the table and on line 1791, but it seems to us it should be a chain ID not a certificate ID, since *cid* is depicted as a decimal parameter that follows "/chain/" in queries.
23. FIPS 140-2 is referenced [line 2132] but should now be FIPS 140-3, which NIST formally approved earlier this year.[8]
24. On line 530, "characters strings" should say "character strings".

Thank you for this opportunity to provide feedback on NISTIR 8213.

Sincerely,

Michael McCormick

Michael McCormick
President, Taproot Security
www.taprootsecurity.com
mike@taprootsecurity.com

[7] https://www.nist.gov/programs-projects/nist-randomness-Beacon
[8] https://csrc.nist.gov/Projects/FIPS-140-3-Development