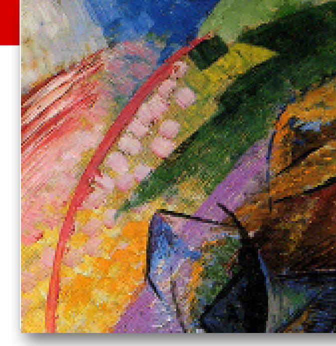
the information security provider

How random is your random? Assessing Entropy with SP800-90B

Stephan Mueller <Stephan.Mueller@atsec.com>



Agenda

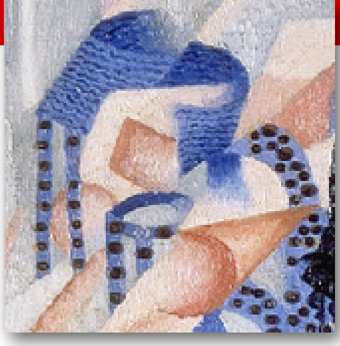
- SP800-90B: Nice formulas, but what do I do with them?
  - Test approach suggestion
  - Example
- Oh joy, I have results out of the SP800-90B formulas – what do the numbers mean?
  - Interpretation help

SP800-90B Tool

- SP800-90B tool available at https://github.com/usnistgov/SP800-90B_EntropyAssessment.git
- Binary Input Required
- Decision between IID and non-IID
- “Block Size” limited: Markov Test uses at most 6 bits
- → My raw noise is not a bit stream – what shall I do?
- → My raw noise is a time stamp – how to handle?

SP800-90B Tool: Input Data Format

- Block size: the width of a data block that is generated in a dependent fashion
- Binary data (e.g. ring oscillator)
  - Block size <= 6 bits? → Simply process with tool!
  - Block size > 6 bits? → Take 6 fast moving bits out of each block and concatenate to form bit string
- Integers: Counters, Register values
  - Take at most 6 bits of fastest moving part → form binary string by concatenation
  - Example: Linux /dev/random noise source: high-resolution time stamp
    - Take 4 or 6 least significant bits of time stamp and concatenate



IID or non-IID

- Do blocks of noise data have dependencies?
  - This question can often be answered easily. The most likely answer is: they are non-IID.
  - If you cannot answer it, assume they are non-IID.
  - Only apply the IID case if there is a valid rationale.

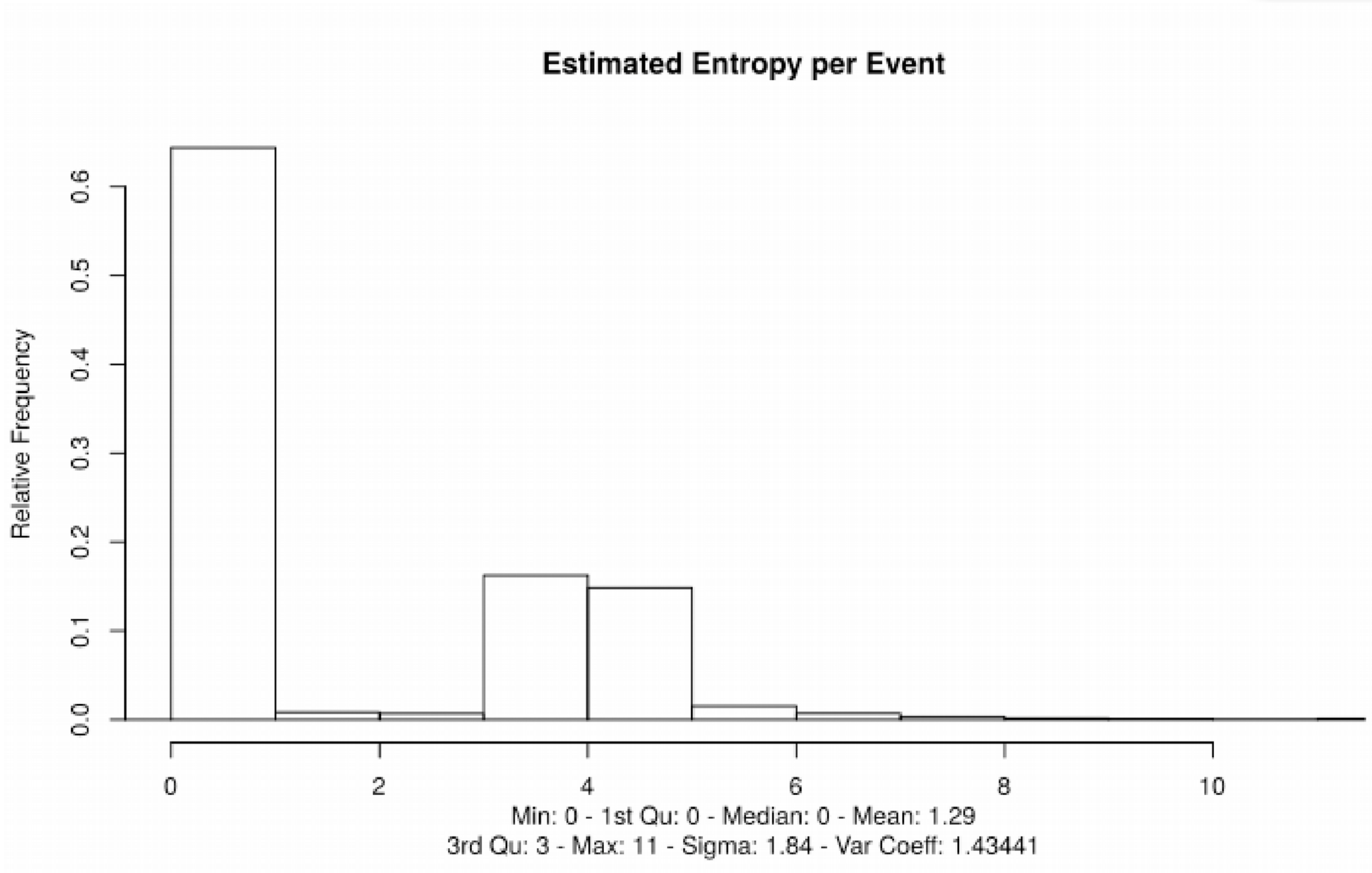
SP800-90B Results

- The SP800-90B tool returns some minimum entropy value.
  - This value is relative to the block size.
- Example Linux interrupt noise source's high-resolution time stamp:
  - 32 Bit value for each interrupt → collect 1,000,000 samples
  - Take 4 least significant bits from each time stamp → bit stream
  - Process bit stream with SP800-90B tool
  - Tool result: 1.97961 (bits)
  - → 1.97961 bits of entropy per 4 data bits
  - → Make life easy – worst case applied: 1.97961 bits per time stamp (per data block)

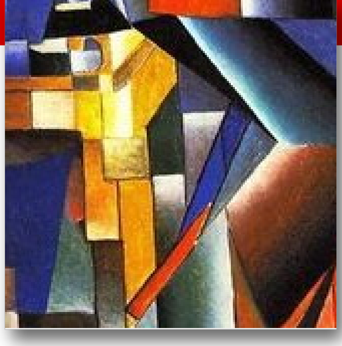
Tool Result Interpretation

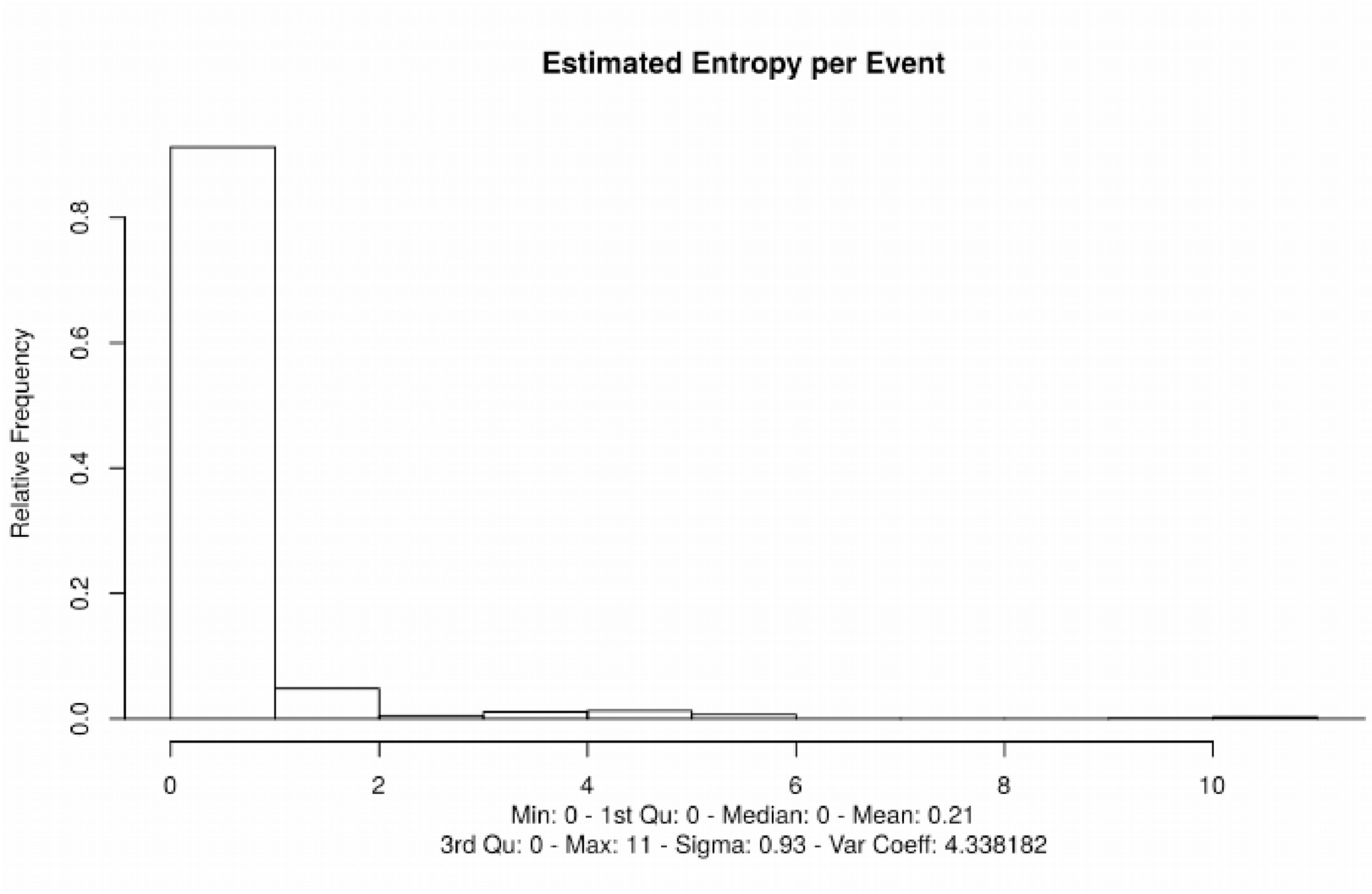
- The entropy value from tool must be compared with entropy implied in use case!
- The DRNG entropy requirement divided by the tool entropy content result equals the minimum seed size from noise source.
  - E.g. binary data results in 0.5 bits of entropy → seed size is twice the entropy requirement
- If entropy heuristic is in place, compare heuristic value with tool entropy content value:
  - Linux Interrupts: 1 bit of entropy per 64 interrupts (i.e. 64 time stamps) – tool indicates each time stamp has 1.98 bits of entropy
  - Linux HID/disk events: compare average heuristic entropy value with tool result
    - Linux HID event tool result: 1.889 bits per time stamp
    - Linux disk event tool result: 2.72828 bits per time stamp

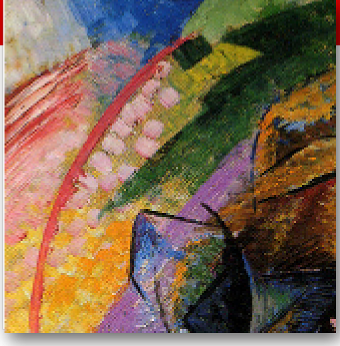
Linux HID Heuristic Entropy Estimate



Linux Disk Heuristic Entropy Estimate







Conclusion for Linux RNG

- Comparing of obtained data
- Heuristic entropy is always smaller than measured entropy
- → Linux RNG underestimates entropy
- → Linux RNG is conservative
- → Linux RNG entropy estimation guarantees that the stated amount of entropy is really present in entropy pools
- → Data / entropy ratio out of /dev/random is almost 1:1
- → getrandom syscall delivers at least 128 bits of entropy

Noise Source	Heuristic Entropy	Measured Entropy
HID	1.29	1.89
Disk	0.21	2.73
IRQ	<= 1/64	1.98