WHITEWOOD

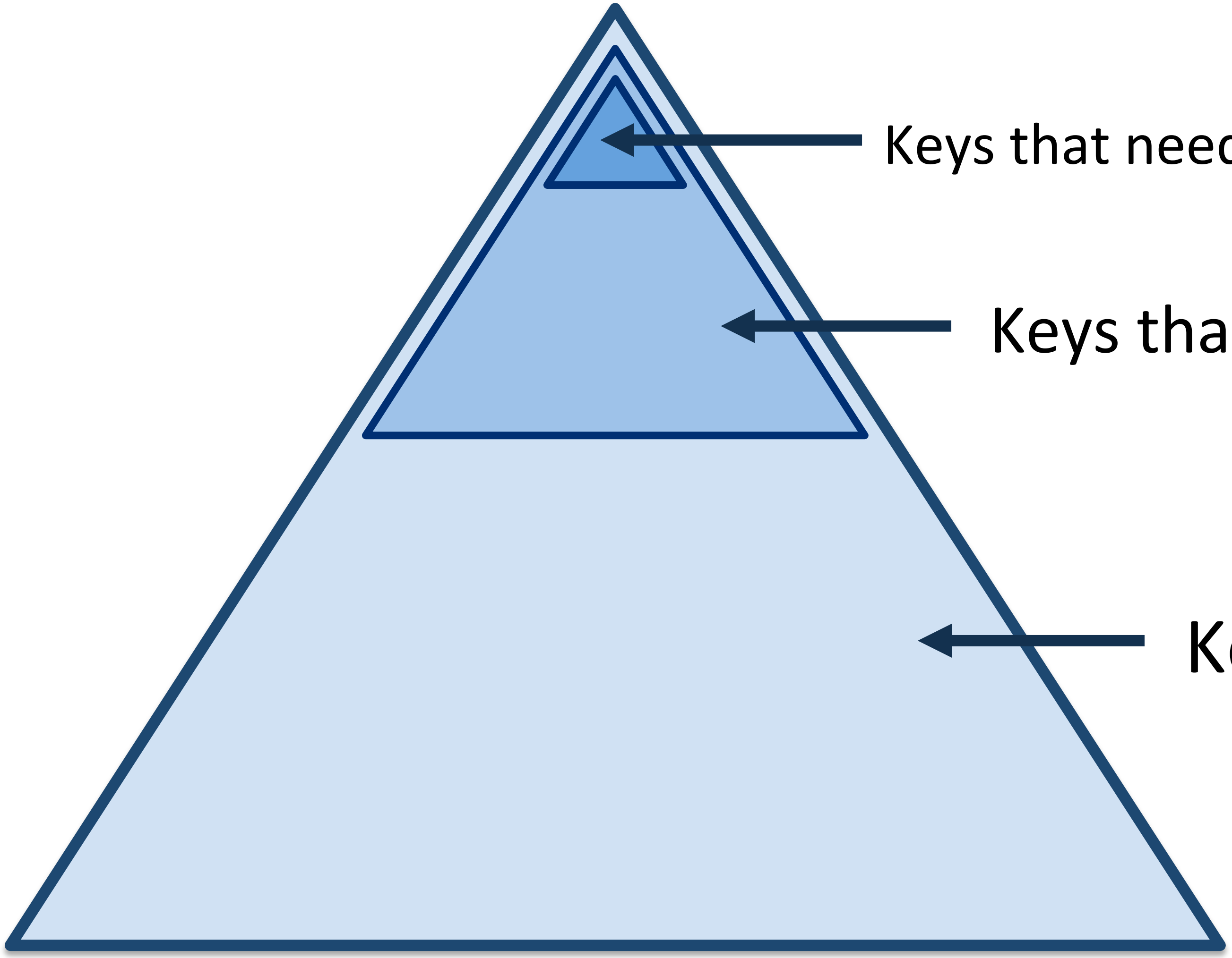
Entropy – A Case of Supply and Demand

ICMC 2017 – Washington DC

Richard Moulds – General Manager, Whitewood

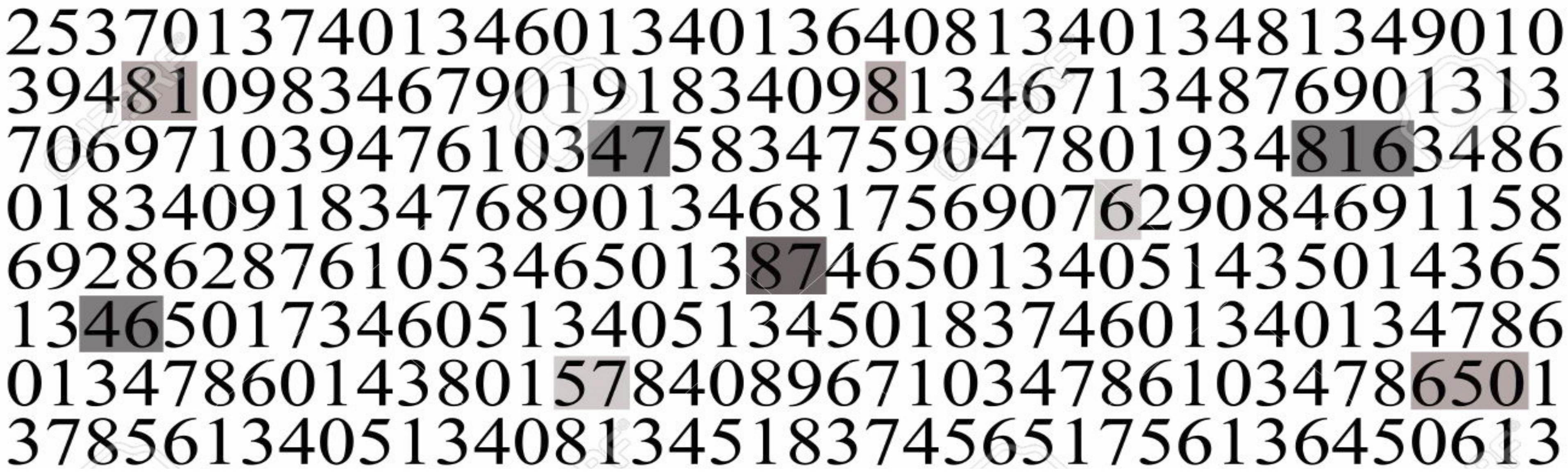
May 19th 2017

Keys to the kingdom



WHITEWOOD

All crypto security starts with random numbers



Crypto security assumptions rely on keys being random, when patterns emerge (or are engineered) keys get more predictable

Anything less than true randomness is a risk

WHITEWOOD

But, there’s a problem

We need more and more randomness

But, it’s getting harder to be truly random

- More and more crypto in use
- Longer and longer keys
- Increased key management scrutiny
- Tougher compliance
- Quantum threat…

- Abstraction, containers and VMs
- Hosted and cloud environments
- Headless systems, no ‘users’
- Snap shots and replication
- Low power IoT devices

WHITEWOOD

Hidden vulnerabilities - backdoors of choice

WHITEWOOD

Basic requirements for randomness

Uniformity: As many 1s as 0s, on average

Independence: Each bit uncorrelated with all previous - statistical tests

- Diehard(er), NIST SP800-22 STS, TestU01 etc.

These are necessary, but are **not** sufficient:

- π passes these tests –

  3.141592653589793238462643383279502884197169399375896

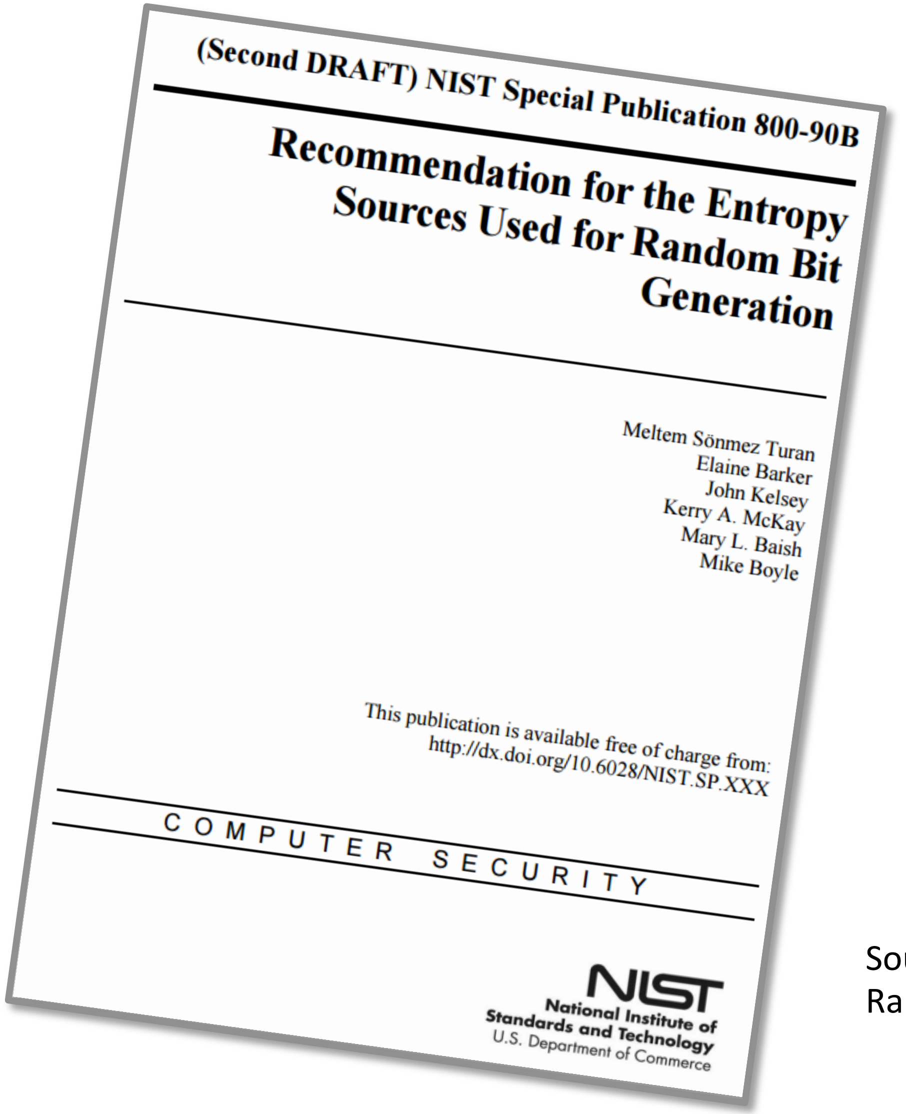
For cryptography, we also need -

- Unpredictability – forward and backward protection
- Imperturbability
- Secrecy
- Reliability

Proving true randomness requires knowledge of the source of randomness, not just statistical analysis of the output

WHITEWOOD

Finally we have a standard (nearly)



(Second DRAFT) NIST Special Publication 800-90B

Recommendation for the Entropy Sources Used for Random Bit Generation

Meltem Sönmez Turan
Elaine Barker
John Kelsey
Kerry A. McKay
Mary L. Baish
Mike Boyle

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.SP.XXX

COMPUTER SECURITY

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

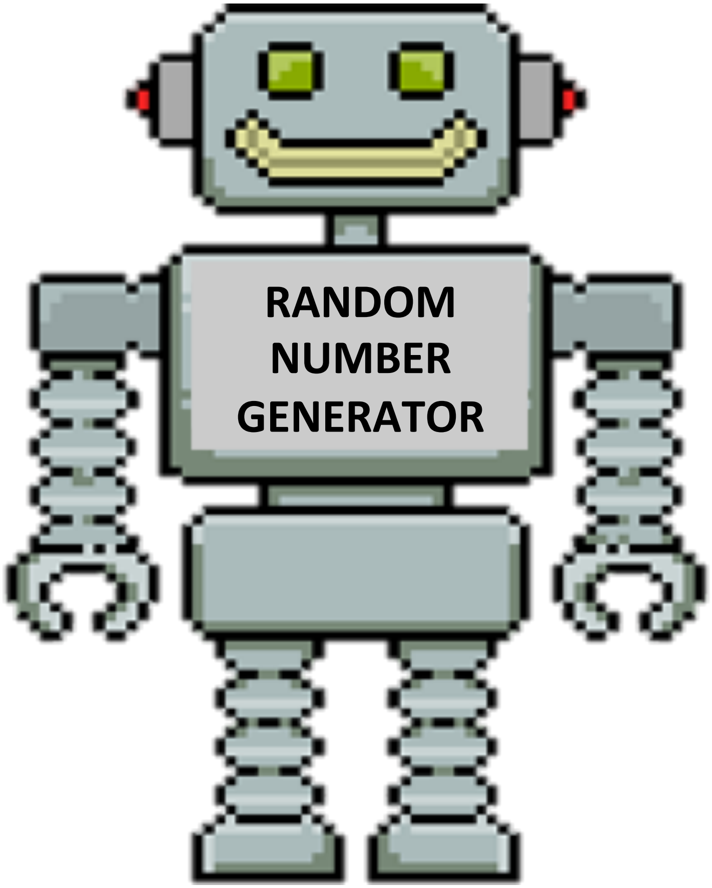
"Specifying an entropy source is a complicated matter. This is partly due to confusion in the meaning of entropy, and partly due to the fact that, while other parts of an RBG design are strictly algorithmic, entropy sources depend on physical processes that may vary from one instance of a source to another".

Source – Recommendation for the Entropy Sources Used for Random Bit Generation (SP800-90B 2nd draft) – NIST January 2016

WHITEWOOD

Why so complicated?

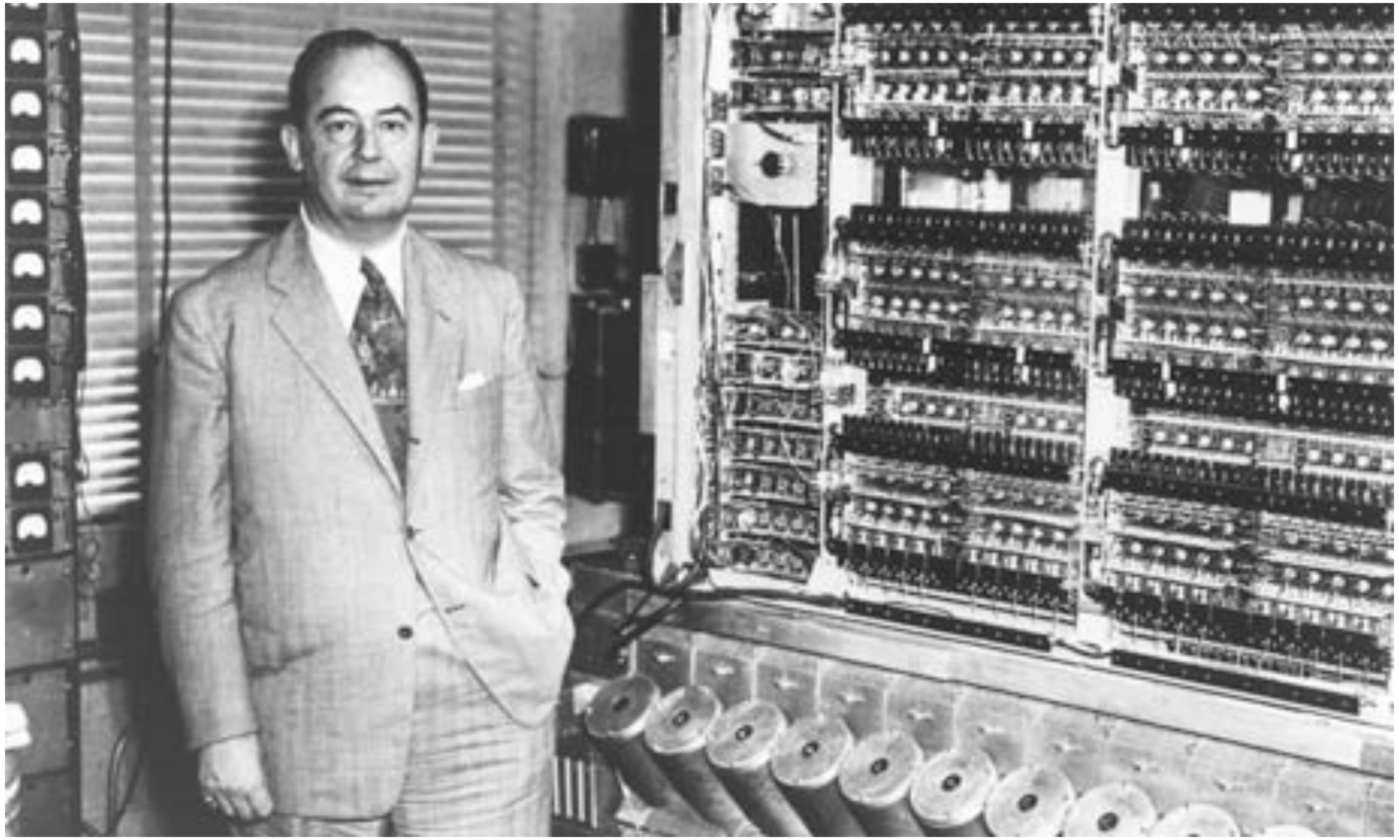
Most random numbers come from the Operating System



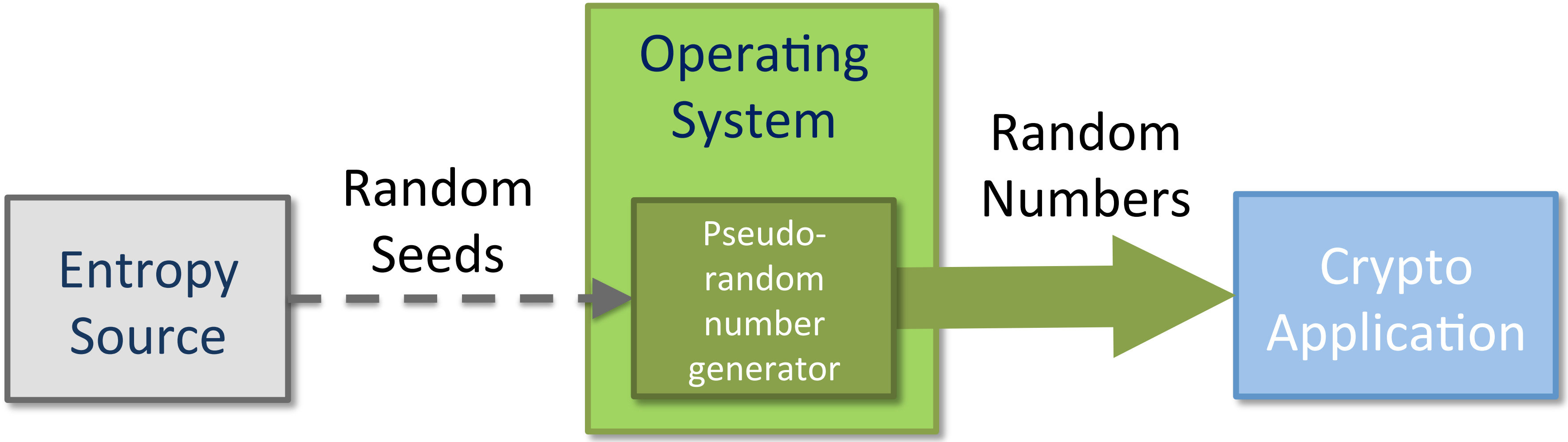
But software doesn't act randomly

Entropy - a long standing issue

"Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin." (J. von Neumann, 1951)



Pseudo-random numbers – an oxymoron?



Shuffling the deck

Dealing the deck

WHITEWOOD

PRNG assumptions

- PRNGs are well defined and easily validated (SP800-90A)
- In principle need very minimal seeding

	SHA-1	SHA-224 and SHA-512/224	SHA-256 and SHA-512/256	SHA-384	SHA-512
Maximum entropy input length (*max_length*)	$\leq 2^{35}$ bits				
Seed length (*seedlen*) for Hash_DRBG	440	440	440	888	888
Maximum personalization string length (*max_personalization_string_length*)	$\leq 2^{35}$ bits				
Maximum additional_input length (*max_additional_input_length*)	$\leq 2^{35}$ bits				
max_number_of_bits_per_request	$\leq 2^{19}$ bits				
Number of requests between reseeds (*reseed_interval*)	$\leq 2^{48}$				

Source - NIST SP800-90A

Security assumptions

- Sufficient amount of entropy
- Sufficient quality of entropy
- Secrecy of seed values
- Secrecy of internal state
- Independence of internal state across multiple instances

Lack of confidence in assumptions should force up the reseeding rate

WHITEWOOD

Entropy supply and demand - CISO’s reality

Payment terminals

User devices

vmware

Private datacenter

docker

openstack™

Private clouds

amazon web services

Microsoft Azure

Public clouds

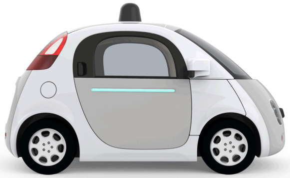
Google Cloud Platform

Network infrastructure

CISO’s Confidence Level?

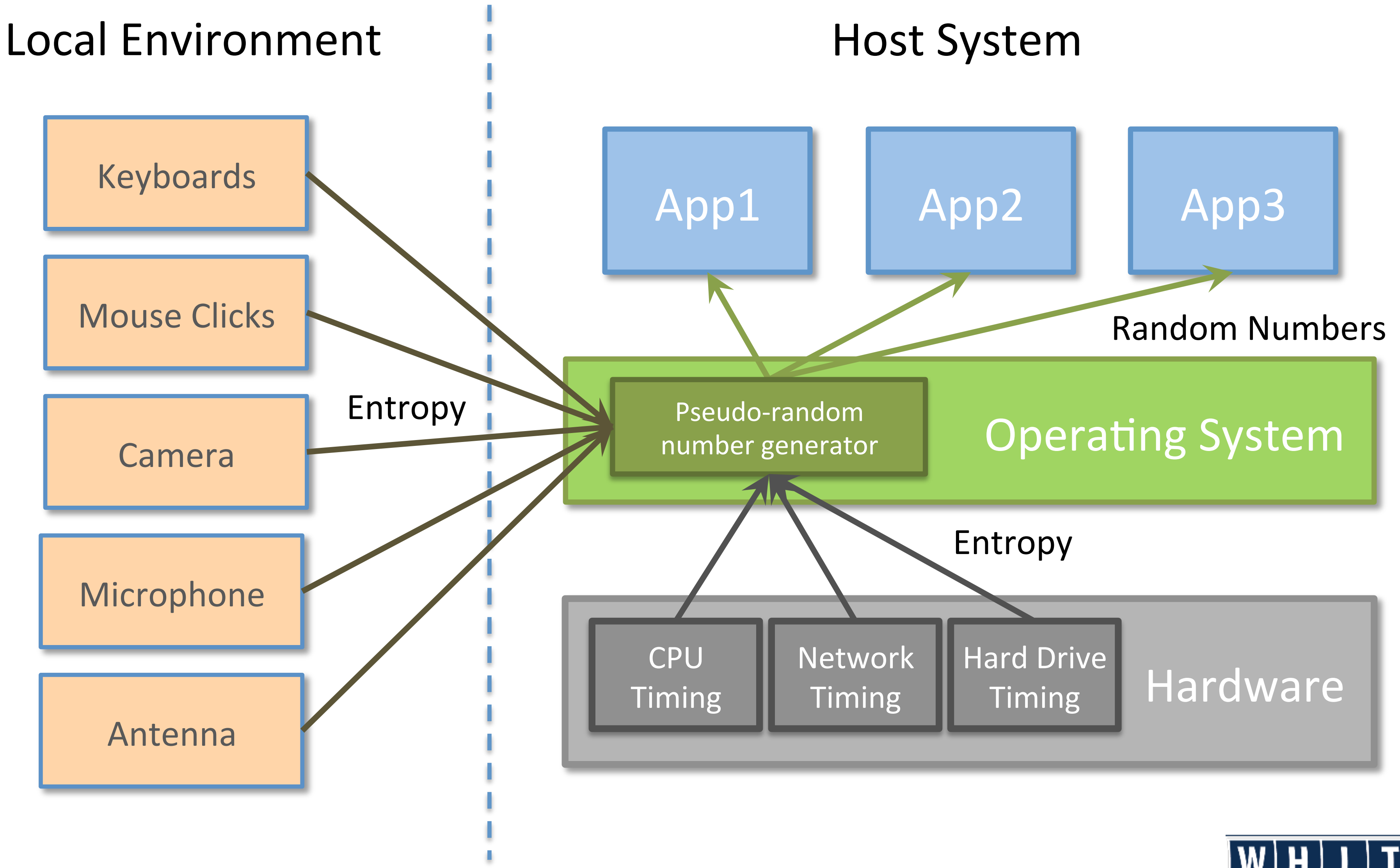
- Quantity
- Quality
- Secrecy
- independence

IoT devices



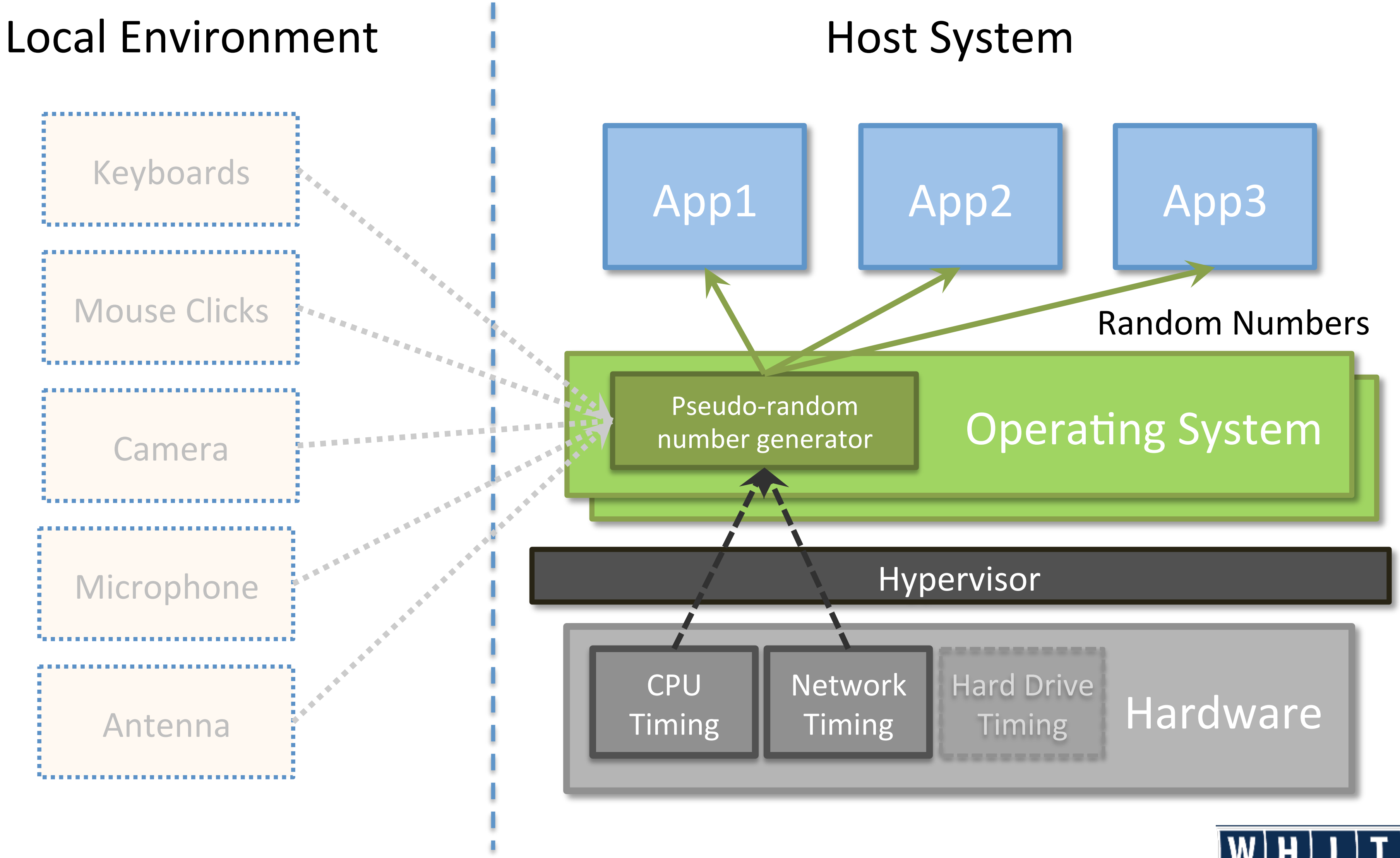
WHITEWOOD

Where does entropy come from?



WHITEWOOD

But in a virtual world…

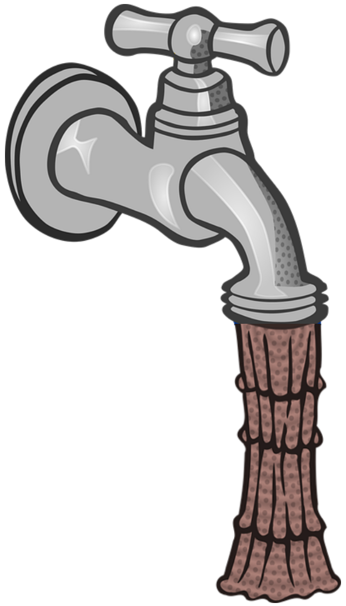


Random number generators in Linux

dev/random

Delivers random numbers only if sufficient entropy has been captured - otherwise it stops

dev/urandom



Delivers random numbers irrespective of how much entropy has been captured

WHITEWOOD

Interrupt derived entropy in Linux

Kernel IRQ handler adds data from interrupts into the Entropy Pool

Cycle Count & Kernel Timer (4 bytes)	IRQ (4 bytes)	Instruction Pointer (8 bytes)

Cycles	Kernel	IRQ	Instruction Pointer
123975895488	4294893898	14	18446744071578900000
123977123888	4294893898	14	18446744071578900000
123979445304	4294893898	14	18446744071578900000
123983781984	4294893899	14	18446744071578900000
123985083096	4294893899	14	18446744071578900000
123986825584	4294893899	14	18446744071578900000
123987250920	4294893899	14	18446744071578900000

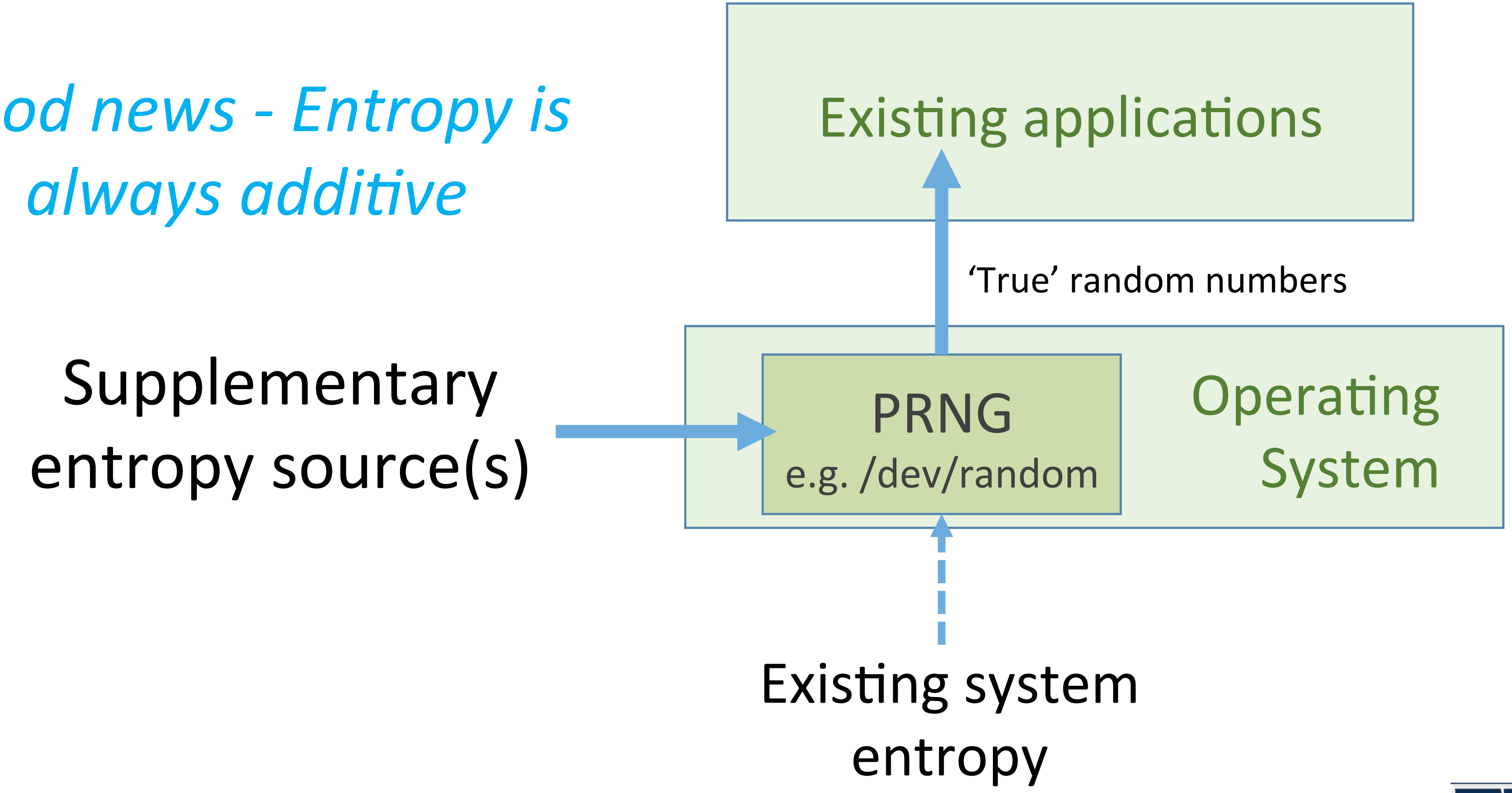
Thanks to Adam Everspaugh - http://pages.cs.wisc.edu/~ace/

WHITEWOOD

Enhancing system randomness

Goal: generate true random numbers from a PRNG

Good news - Entropy is always additive



Supplementary sources of entropy

Three general approaches to improve entropy:

1. Software daemons to more efficiently extract entropy from existing signals and interrupts
   - State changes - HAVEGED (www.issihosts.com/haveged/)
   - Timing Jitter – CPU Jitter RNG (www.chronox.de/jent.html)
   - Microphones and cameras – ‘noise’ sources (www.vanheusden.com/aed/ and www.lavarnd.org/)
2. Local hardware based entropy sources
   - Embedded CPU feature - e.g. Intel RdRand
   - Plug-in devices - USB sticks, HSMs, PCI cards, etc.
3. Network based services for entropy sources and RNGs
   - Random number services (https://random.org and https://beacon.nist.gov/home)
   - “Entropy as a Service” (https://getnetrandom.com and https://entropy.ubuntu.com/)

Comparison of supplementary entropy sources

	Entropy Extractors	Embedded Hardware Sources	Retrofit Hardware Sources	Entropy as a Service
Form factor	Software	Platform hardware	Plug-in hardware	Service (public or private)
Advantages	• Low cost (free) • Lots of choices • Open source	• Low cost (free?) • Convenient • Speed	• High assurance • Speed	• Consistency • Scale • Assurance • Low cost (free?)
Barriers	• Hard to validate • Hard to Manage • Open source	• Platform specific • Hard to validate • Requires configuration	• Inconvenient • Not cloud friendly • Cost	• Immaturity • Complexity (?)

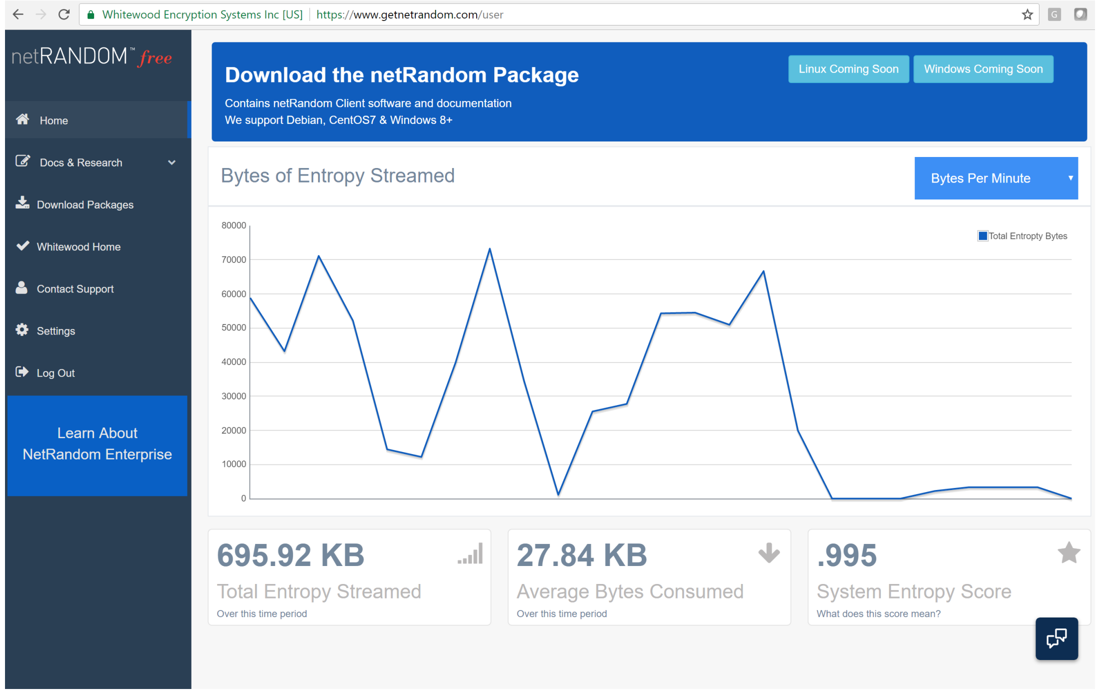
WHITEWOOD

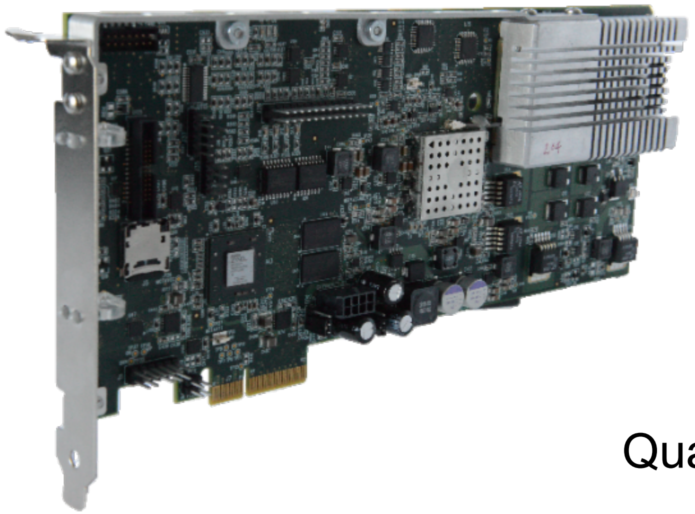
Shameless plug

netRANDOM™

Network Delivered Quantum Entropy
(Entropy as a Service)

- netRandom Free – cloud service - getnetrandom.com
- netRandom Enterprise - private, on-premise entropy servers





EntropyEngine™

Quantum Random Number Generator

WHITEWOOD

Summary

- Random numbers are critical for security but are often poorly understood and managed
- Random number generators are a point of attack and vulnerability – potentially an invisible one
- Modern application environments present entropy challenges – VMs, cloud and IoT
- Testing the quality of entropy sources and random number generators is hard
  – new standards will help (NIST SP 800-90B)
- Supplementary sources of entropy improve security and there are plenty of deployment options
- Random number generation should be a critical component of your key management strategy and datacenter infrastructure planning

WHITEWOOD

Thank you

richard.moulds@whitewoodsecurity.com

Entropy-as-a-Service at www.getnetrandom.com
Demo at www.whitewoodsecurity.com/netrandom-demo