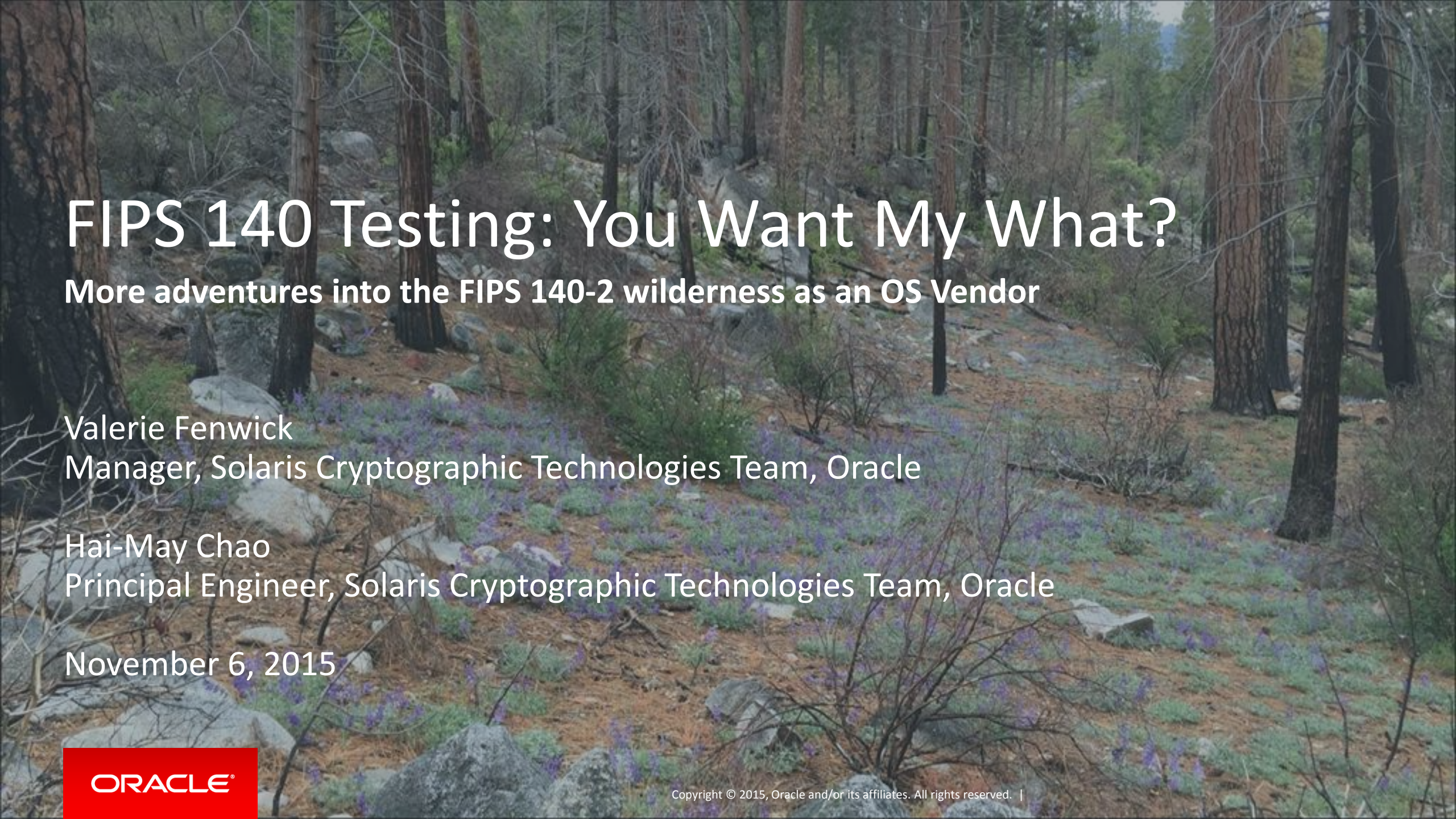

ORACLE®



FIPS 140 Testing: You Want My What?

More adventures into the FIPS 140-2 wilderness as an OS Vendor

Valerie Fenwick
Manager, Solaris Cryptographic Technologies Team, Oracle

Hai-May Chao
Principal Engineer, Solaris Cryptographic Technologies Team, Oracle

November 6, 2015

ORACLE

Terminology…Still

- Consultants ask for “public API”
- Engineers panic!
- Huh?

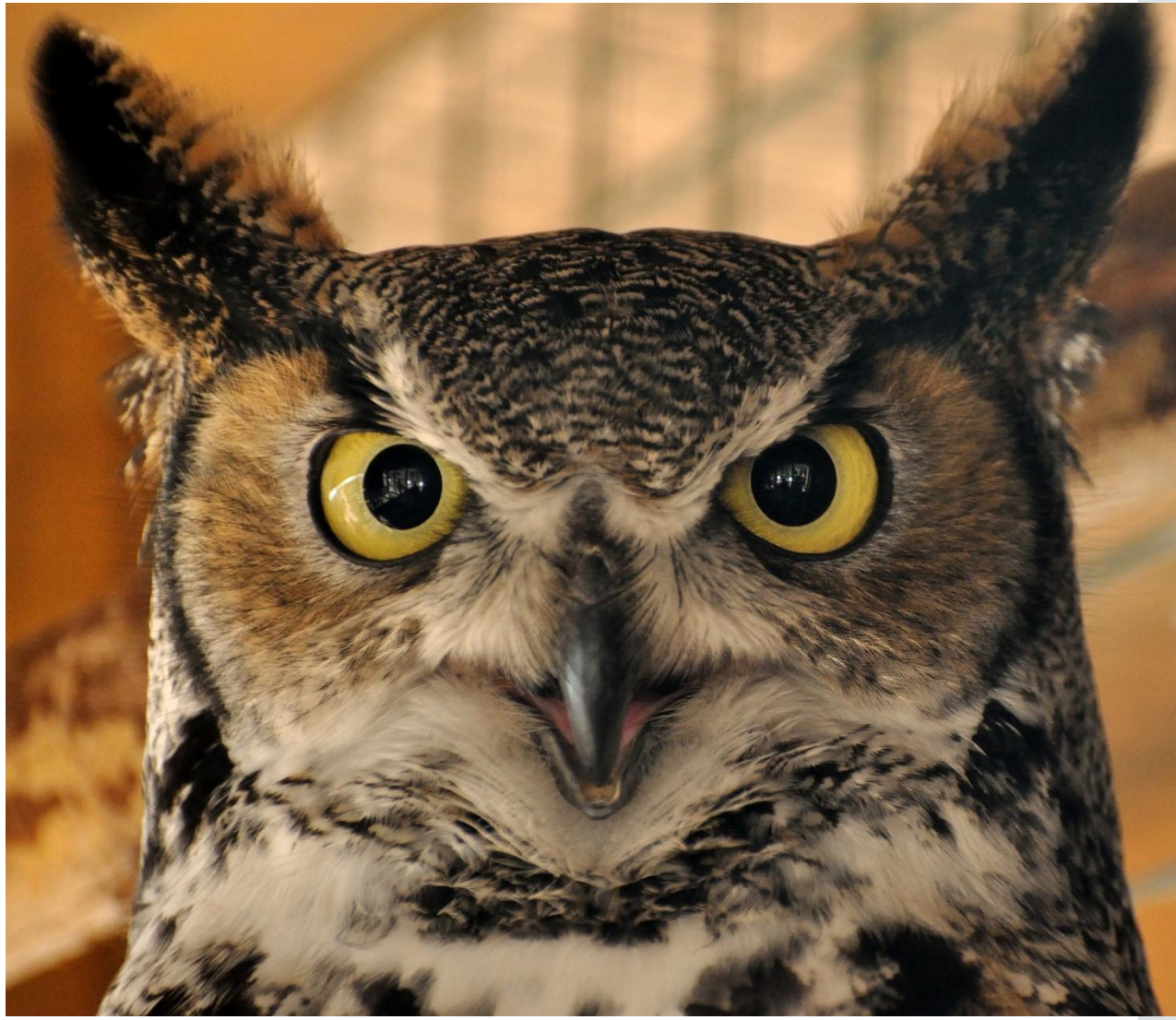


Photo by Mary Shattock, Creative Commons

Huh?

- Consultants want something they can call from their test programs
  - Something defined as a part of the system, in the header files
- To an engineer, “Public” means something we document, stringently test, allow all of our customers to use, and fully support
  - And … quite frankly, we don’t want our customers using **THESE** interfaces!

Background

Our gripes specific to us?

- The same code is compiled for our userland libraries and our kernel modules
  - Creates multiple different software modules
    - NOTE: We are NOT talking about cryptographic modules, we're talking kernel modules and user level libraries. (Terminology)
  - **EXACT** same source code
  - APIs entry points vary
    - Public PKCS#11 and private ucrypto API in userland
    - Proprietary private (PKCS#11-like) kCF API in kernel
  - Due to the way FIPS 140-2 is defined we get **TWO boundaries**

That's Great – Who Cares?

Really – Same Source Code

- Can't use DRBG from other boundary
  - But, it's the same code!!
- Must have two copies of DRBG
  - Necessitates duplicating code here

But Wait, There's More

Call now…

- Solaris has at least **THREE** cryptographic FIPS 140-2 boundaries
  - OpenSSL
    - It's another API our customers expect in an OS
  - Sometimes, for fun, our customers use NSS, too

Can’t We Share?

Apparently not

- Many of these boundaries look at the same configuration files.
  - Often it’s the same source!
- Yet, we cannot call an approved DRBG from another boundary.
- As an OS, we supply getrandom(2) system call.
  - getentropy(2), too!
- But, we all have to reinvent the wheel.

If It's Good For the Consultant….

- Consultants ask for strange things

Photo by Tambako The Jaguar, Creative Commons

“Special” DSA API

- Test harness has to be able to supply the "random" number, so you can generate a known signature.
- **BUT**, this is essentially a back door.
  - But there’s no other way…
- We **CANNOT** ship this in a production release
  - Can’t even put it in the code as debug – one tiny coding error, and **BANG** – back door!
- So…. We make a special version of our binaries for consultant

“Special” RSA API

- Key generation changes to print out **keys** and **primes**
- Signature generation and verification using the provided messages
- Nope, can’t include this in our production code, either
  - Not even as “**DEBUG**” code – too easy for one programming error to make it live.
- Required code changes complex
  - All callers (in private copy) MUST be changed.
  - Significant Engineering effort

Again, WHO CARES?

Our Dirty Little Secret

- The binaries tested are ***NOT*** the ones we ship to customers.
- Consultants do not test what the customer gets.
- **But** the standard is written so that what we test is supposed to be what we ship.
- We cannot put this code in the repository
  - Just not safe, so making special test builds
- Getting less and less like the real deployment.

We Still Have Integrity!

Um…

- For the labs:
  - We have to build the modules with wrong HMAC value, so integrity check will fail
  - RNG generator tests
    - We specify a fixed random, and verify the result.
    - Needed to provide the wrong seeding value to cause it to fail.
  - More special versions of our binaries
- And… we don't even list the expected hash of the modules so they can verify they have what we actually validated.
  - Which is what, exactly?

But We Can Keep Doing This, Right?

Wrong

- CPUs and OSes are advancing much faster than the FIPS 140 standard
- Soon, CPUs and OSes will not allow us to do this malarkey
  - Memory protection improving every year
  - Will simply refuse to load the tainted binary

How Does This Impact Our Customers?

- What our consultants and labs are testing is simply NOT what we ship to the customer.
- Yet, we're told we can't even provide non cryptographically relevant security vulnerability fixes without a Change Letter (**$$$**)
  - Like returning the wrong error code, spelling error or non-keying material memory leak.
- "We trusted vendors once and they lied"
  - You trust us to make these failure demonstration modules…

Implementation “Guidance”

- NOT Guidance – **Requirements**
- Between Solaris 11.1 and 11.3, we just needed to meet a *few* new pieces of implementation guidance …

These Are **JUST** Clarifications, Right?

Sadly, No.

- To address all of the new IG over the last 3 years:
  - Nearly 100 distinct code changes required
  - **New** DRBG: SP 800-90A
  - Update from FIPS 186-3 to **FIPS 186-4**
  - IG requiring change to how “fips mode” is set
    - Now install option

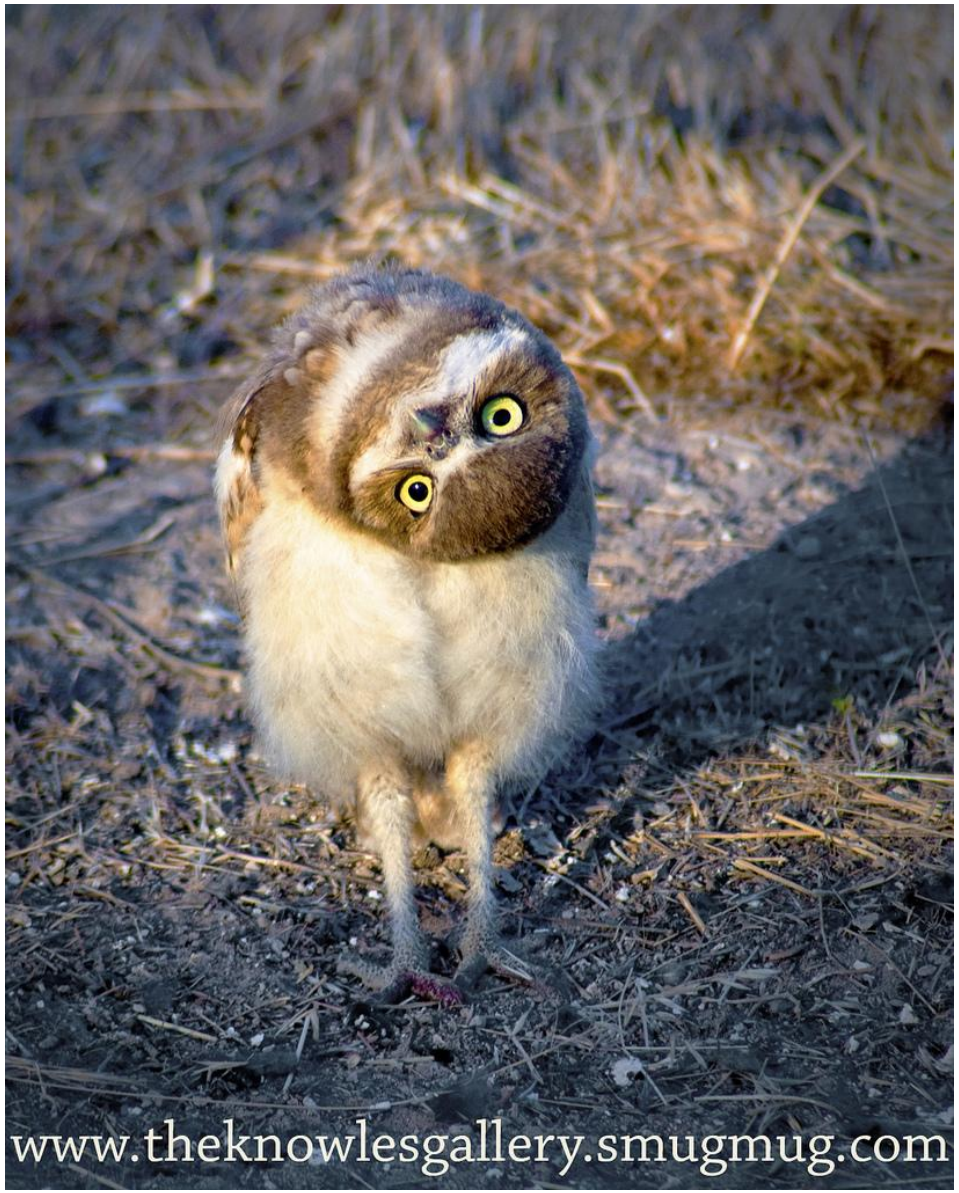


Photo by Charles Knowles, Creative Commons

Clarifications? Really?

- NIST now making big IG changes with no grace period
  - IG A.5 AES-GCM IV Generation
    - Our release was “in the can”, and had to have firedrill to fix not just our code, but consumers as well (ZFS file system)
    - The standard PKCS#11 v2.40, released last year, is NOT compliant
      - So, ***NO*** PKCS#11 APIs from ***any*** vendor will be able to pass this CMVP requirement.
      - PKCS#11 TC will have to move to v3.0
  - Recall, not all FIPS boundaries are at the hardware level, some are part of a bigger system - like an OS!

TANSTAAFL

- Firedrills cost vendors money and time
  - Consultants may have already completed algorithm testing, submitted to lab
  - May have already received CAVP certificates
  - Contracts may have to be rewritten
  - Delays getting validated products into the hands of customers.

Photo by gerry, Creative Commons

Sunsets

- Sunsetting historical FIPS 140-2 validations
  - Lack of SP 800-90A compliance
    - Which… didn’t exist when those products were validated
    - Those old validations were 100% correct at the time
    - You've EOLed those certificates without revving the standard.
      - Yet, we all have a FIPS 140-2 validations.
    - Why is that okay?

Our Customers

FIPS 140-2 Now Meaningless

- How is a customer supposed to know the difference between a product validated under FIPS 140-2 in 2012 and another under FIPS 140-2 in 2015?
  - The certificates look the same
  - Security policies don’t mention which IGs had recently changed
  - We all know those products will have to be significantly different

What Does This Mean?

- Can’t even tell our customers:
  - “Hey, we validated to comply with IG 1.2, 2.3, A.1”
    - Because the IG content changes, but the IG number stays the same
  - Customers have to
    1. Know and understand all IGs
    2. Know when each IG was updated and what that means
    3. Know when the vendor submitted to CMVP
    4. Sometimes, know when they completed CMVP
    5. Cross reference to know what they’re getting

Honesty

Is Such a Lonely Word

- If every vendor was doing the same thing, but not the thing NIST wanted:
  - It's not obvious
  - It's not a "clarification"
- The latest Implementation Guidance updates are truly ***new*** revisions
  - Let's do "minor" revs: FIPS 140-2.1, FIPS 140-2.2

Photo by merec0, Creative Commons