

Build FIPS 140 into your storage products

CIQ

Jason Rodriguez (CIQ)

Jeremy Allison (CIQ/Samba Team)

What is FIPS ?

The Federal Information Processing Standards (FIPS) of the United States are **a set of publicly announced standards** that the National Institute of Standards and Technology (NIST) has developed **for use in computer systems** of non-military, American government agencies and contractors.

● FIPS 140-2

Mandatory standard for the protection of sensitive or valuable data within Federal systems

April 1, 2022 - Cryptographic Module Validation Program (CMVP) no longer accepts FIPS 140-2 submissions for new validation certificates.

September 22, 2026 - All FIPS 140-2 certificates are placed on the Historical List

- **Historical List** - This does not mean that the overall FIPS-140 certificates for these modules have been revoked, rather it indicates that the certificates and the documentation posted with them are more than 5 years old and have not been updated to reflect latest guidance and/or transitions, and may not accurately reflect how the module can be used in FIPS mode

● FIPS 140-3

FIPS 140-3 is an incremental advancement of FIPS 140-2. More restrictions on allowed algorithms and key sizes.

Organizations regulating FIPS







- National Institute of Standards and Technology (NIST)
 - Cryptographic Algorithm Validation Program (CAVP)
 - Cryptographic Module Validation Program (CMVP)

The FIPS 140 standards documents are written by and for experts in cryptography, not mortal storage engineers. The most common comment from storage programmers (me) encountering FIPS standards for the first time is: *“But what does that mean ?”*

Cryptographic Algorithm Validation Program - CAVP

- Ensure algorithms meets NIST standards
- Algorithms testing is conducted by injecting known inputs to ensure algorithms function correctly
- Once testing has completed the CAVP will issue a public certificate
 - The certificate validates the algorithm is compliant on a specific Operating Environment (OE)
 - Note – running on a *different* Operating Environment means the certification is no longer valid.

CAVP – Examples of an Operating Environment

Operating Environment	Algorithm Capabilities
<p>Rocky Linux 8 on Intel Xeon E3 (Kaby Lake) </p> <p>Processor: Intel Xeon E3 (Kaby Lake) Operating System: Rocky Linux 8</p>	<p><u>AES-CBC</u> </p> <p>Direction: Decrypt, Encrypt Key Length: 128, 192, 256</p>
<p>Rocky Linux 8 on Intel Xeon E3 (Kaby Lake) </p> <p>Processor: Intel Xeon E3 (Kaby Lake) Operating System: Rocky Linux 8</p>	<p><u>AES-CCM</u> </p> <p>Key Length: 128, 256 Tag Length: 32, 48, 64, 80, 96, 112, 128 IV Length: 56, 64, 72, 80, 88, 96, 104 Payload Length: 0-256 Increment 8 AAD Length: 0, 256, 65536</p>
<p>Rocky Linux 8 on Intel Xeon E3 (Kaby Lake) </p> <p>Processor: Intel Xeon E3 (Kaby Lake) Operating System: Rocky Linux 8</p>	<p><u>AES-CFB8</u> </p> <p>Direction: Decrypt, Encrypt Key Length: 128, 192, 256</p>

Cryptographic Module Validation Program - CMVP

- The module must have the following features
 - Known Answer Tests (KAT)s
 - Executed on startup or module load to ensure the health of the library algorithms
 - Pairwise Consistency Tests (PCT)s
 - PCTs are run against a newly generated or imported keys
 - The following checks are performed depending on the the algorithms capabilities
 - Public Key check
 - Sign/Verify check
 - Encryption/Decryption check
 - Integrity tests
 - A hash of the module on disk is generated and compared against a known answer
 - Ensure that any unapproved algorithm features are not executed

Cryptographic Module Validation Program (CMVP)- Lab validation

● Functional testing

- Code is annotated with print statements to show each KAT/PCT runs successfully.
- Code is modified to inject failures into KATs/PCTs for on demand failures.
- Demonstrate that unapproved algorithms return an error on execution.
- OpenSSL is the only library with published functional test code.
 - Rocky Linux is planning to change this.

● Source code review

- A source code review form is filled out for the lab.
- Source code review is conducted.

● Security Policy

- The security policy documents all approved/unapproved modes of operation.

FIPS packages in Rocky Linux

FIPS Rocky 8.6

- NSS 3.79.0
- Libgcrypt 1.10.0
 - Rolled forward from 1.8.5-7
- Kernel 4.18.0-425.13.1
 - Rolled forward from 4.18.0-372.32.1
- Openssl 1.1.1
 - Upstream EOL - 09/11/2023
- GnuTLS 3.6.16
 - Backported

FIPS Rocky 9.2

- NSS 3.79.0
- Libgcrypt 1.10.0
- Kernel 5.14.0-284-11.1
- Openssl 3.x
 - 3.x was redesigned for FIPs 140-3
- GnuTLS 3.7.6

FIPS Certified vs. FIPS Compliant.

- A FIPS Certified product has gone through the entire certification process described in previous slides, and has been submitted to a NIST-approved laboratory and passed.
- It is being run on the certified operating environment.
 - Run this code on a different motherboard, make one change to the BIOS (turn on Intel AESNI instructions if certified with AESNI-off, or turn them off if certified with AESNI-on) and the product is no longer certified.
 - Some customers **REQUIRE** this level of assurance. You'll know if you run into one. They'll tell you.
 - Most customers do not.
- A FIPS Compliant product is based on the FIPS certified code, but isn't being run on the certified operating environment.
- Most customers wanting FIPS will be satisfied with FIPS compliant code (Note: my personal opinion).

What are unapproved algorithms ?

Applications that depend on unapproved algorithms can use the algorithm as long as the algorithm is not being used for verification or security reasons (but why else do you need them ? :-).

Partial list of unapproved algorithms

- MD4/MD5
- RC4
- 3DES
- GHASH
- RSA (non-compliant with key sizes below the minimums Approved for FIPS mode and with untested functions)
- HMAC (key size < 112 bits)

Unapproved algorithms and storage protocols

Some storage and authentication protocols haven't kept up with FIPS standard algorithms.

NTLM which uses MD4/MD5.

iSCSI with CHAP which only *requires* MD5. Linux kernel iSCSI initiator can use FIPS approved SHA256 in CHAP, but some vendors only support CHAP with MD5.

SMB3 can be run in a FIPS approved mode. SMB1 is not FIPS approved.

NFS v3/v4 can be run in a FIPS approved mode (no AUTH_SYS).

Some MS-RPC pipes have old crypto embedded (LSA uses RC4).

Building a FIPS Compliant Storage Product

- Note – this slide is not about building a FIPS **Certified** storage product.
 - For that, you'll have to go through the entire process with the approval labs on your own and certify all your own modules.
 - Rocky should make this easier once the test code is all published.
- Start with a FIPS certified Linux (sure, Windows and other OS'es have been FIPS Certified, but this is an Open Source talk).
- All use of crypto algorithms must be done via the FIPS conforming modules (libraries).
 - Modules require Power On Self Test (POST) testing.
 - Known answer tests (KAT), Pairwise Consistency tests (PCT) and module integrity tests (checking hash of library binary matches an on-disk stored value) must be run on library load.
 - Use of algorithms outside of the FIPS conforming modules is **disallowed**.
 - (Unless used inside a channel encrypted using FIPS-approved algorithms)

Building a FIPS Compliant Storage Product (continued)

- Turn on FIPS mode (`fips-mode-setup --enable`) and run regression tests with your own code on top of Linux to see what breaks.
 - If your code uses the “standard” algorithm libraries that have been FIPS certified then if you’ve used unapproved algorithms your code will (correctly) break.
 - Unfortunately this won’t catch every case.
- Some older software ‘hand-crafts’ crypto code – for example, Samba versions before 4.12.
 - Samba 4.12 and later use GNUTLS API’s for all crypto code.
- Because of this you **MUST** audit all use of crypto API calls in your code, and all open source code outside of the crypto libraries that have been FIPS certified.
 - Beware of hand-crafted code that will escape notification of not being FIPS compliant.
- Test Active Directory / Kerberos connections. Kerberos must be hardened to be FIPS compliant.
 - Watch out for “Cryptosystem internal error” messages.

Dealing with unapproved algorithms

- What if you **have** to use unapproved algorithms due to protocol requirements or customer requirements ?
 - Some libraries (gnutls) allow application code to control FIPS modes via API calls.

```
GNUTLS_FIPS140_SET_LAX_MODE();
```

... do unapproved operation..

```
GNUTLS_FIPS140_SET_STRICT_MODE();
```
 - Other libraries will require you to rebuild/install a non-FIPS certified version of one of the FIPS libraries.
 - Use linker magic / static link to ensure your unapproved code links to the non-FIPS library.
- NTLM is the worst offender here.
 - *“Sure, I love kerberos but I have to deploy to customers who only have NTLM / their DNS is broken...”*

Adding FIPS to your storage product – the Linux kernel

- The Linux kernel is set into FIPS mode by ensuring the kernel boot command-line parameter has “fips=1”
 - The internals of the kernel take care of ensuring only FIPS-compliant algorithms are available to userspace.
 - Internally it may still use unapproved algorithms for internal purposes e.g. ghash as part of gcm(aes).
 - These unapproved algorithms are not exposed outside the kernel.
 - Secure boot should be used to prevent any change to the kernel command line.

Working with FIPS code in the Linux kernel.

```
alg = __crypto_alg_lookup(name, (type | test) & ~fips,  
                          (mask | test) & ~fips);  
if (alg) {  
    if (((type | mask) ^ fips) & fips)  
        mask |= fips;  
    mask &= fips;  
  
    if (!crypto_is_larval(alg) &&  
        ((type ^ alg->cra_flags) & mask)) {  
        /* Algorithm is disallowed in FIPS mode. */  
        crypto_mod_put(alg);  
        alg = ERR_PTR(-ENOENT);  
    }  
}
```

- The Linux kernel has a mature and widely supported crypto module system.
 - Unfortunately the code around FIPS support is.. *horrible*.
 - And almost completely uncommented.
 - Budget large amounts of engineering time to make sense of this stuff.

Is adding FIPS going to increase my support burden ?

- Oh YES ! :-).
- Once FIPS mode is enabled in your product, network connections to older systems / non FIPS-compliant systems can break.
 - TLS connections have restrictions as to negotiated algorithms / key lengths under FIPS.
 - TLS is used in a lot of software to provide (appropriately enough) transport-level security.
 - Test all support-critical components (admin and support tools) to ensure customer support can access FIPS systems.
 - Train customer support people on common FIPS interoperability problems and how to identify them.
- FIPS-enabled systems are really only completely happy talking to FIPS-enabled systems.
 - Remember, FIPS-140-3 is more restrictive than FIPS-140-2.

Dealing with patches and updates

- Once you have a FIPS **Certified** product, **ANY** change to the certified Operating Environment will mean the product is no longer FIPS Certified.
 - What about patching CVE's ?
 - FIPS Certified systems must be re-certified after patching.
 - This means FIPS Certified systems don't get patched much :-).
 - Re-validation is a faster process that allows CVE changes in FIPS modules to be re-evaluated quicker than a full certification.
- Patching a FIPS compliant library or the kernel (even for a severe CVE fix) means the code is no longer FIPS compliant.
 - Vendors and customers have to chose between leaving potentially severe vulnerabilities unpatched, or losing FIPS compliance.
- CVE fixes in other components of the operating environment don't affect FIPS compliance.

Questions and comments ?