

FIPS 140-3 and Samba/FreeIPA challenges in RHEL 9

Alexander Bokovoy

Julien Rische

Red Hat // Samba Team



Who are we?

- **Alexander:** Sr. Principal software engineer at Red Hat
 - Samba Team member
 - FreeIPA core developer
- **Julien:** Software engineer at Red Hat
 - MIT Kerberos maintainer in Fedora and Red Hat Enterprise Linux



FIPS 140

This standard specifies the security requirements for a cryptographic module utilized within a security system protecting sensitive information in computer and telecommunication systems (including voice systems) as defined in Section 5131 of the Information Technology Management Reform Act of 1996, (Public Law 104-106) and the Federal Information Security Management Act of 2002 (Public Law 107-347).

FIPS 140-1, first published in 1994, was developed by a government and industry working group. The working group identified requirements for four security levels for cryptographic modules to provide for a wide spectrum of data sensitivity (e.g., low value administrative data, million dollar funds transfers, and life protecting data) and a diversity of application environments (e.g., a guarded facility, an office, and a completely unprotected location). Four security levels were specified for each of eleven requirement areas. Each security level offered an increase in security over the preceding level. These four increasing levels of security allowed cost-effective solutions that were appropriate for different degrees of data sensitivity and different application environments.

In 2001, FIPS 140-2 superseded FIPS 140-1. FIPS 140-2 incorporated changes in applicable standards and technology since the development of FIPS 140-1 as well as changes that were based on comments received from the vendor, laboratory, and user communities. Though the standard was reviewed after 5 years, consensus to move forward was not achieved until publication of the 2012 revision of International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) 19790.

FIPS 140-3 supersedes FIPS140-2. FIPS 140-3 aligns with ISO/IEC 19790:2012(E) and includes modifications of the Annexes that are allowed to CMVP (as a validation authority). The testing for these requirements will be in accordance with ISO/IEC 24759:2017(E), with the modifications, additions or deletions of vendor evidence and testing allowed as a validation authority under paragraph 5.2. Major changes in FIPS 140-3 are limited to the introduction of non-invasive physical requirements.



FIPS 140-2 in RHEL 8

- Crypto modules
- System-wide crypto policy
- Application-level compliance

| Core cryptographic component | Description | FIPS 140-2 cryptographic module |
|------------------------------|--|---|
| OpenSSL | General purpose cryptographic toolkit library which includes TLS and DTLS implementations. | Yes |
| GnuTLS | Cryptographic toolkit which is focused towards a simple to use TLS and DTLS implementation. | Yes |
| NSS | The cryptographic toolkit library of the Firefox browser; it follows the Firefox Extended Support Release (ESR) lifecycle with asynchronous updates and feature enablement or removal. | Yes |
| libgcrypt | The GnuPG cryptographic library. | Yes |
| kernel | The Linux kernel internal cryptographic library. | Yes |
| OpenSSH | The SSH client and server applications of the operating system. It depends on OpenSSL for cryptography. | No; It no longer implements FIPS 140-2 relevant cryptography and depends on the OpenSSL module. |
| libssh | A secure communications library implementing the SSH protocol. It depends on OpenSSL for cryptography. | No; It does not implement FIPS 140-2 relevant cryptography and depends on the OpenSSL module. |
| Libreswan | The IPsec client and server applications of the operating system. It depends on NSS and kernel for cryptography. | No; It no longer implements FIPS 140-2 relevant cryptography and depends on NSS module. |

FIPS 140-3 in RHEL 9

- [Cryptographic module validation program](#)
 - Implementation under test

| Module name | Start Date |
|--|------------|
| Red Hat Enterprise Linux 9 libgcrypt | 6/15/2022 |
| Red Hat Enterprise Linux 9 gnutls | 6/15/2022 |
| Red Hat Enterprise Linux 9 kernel | 6/15/2022 |
| Red Hat Enterprise Linux 9 nss | 6/15/2022 |
| Red Hat Enterprise Linux 9 OpenSSL FIPS Provider | 6/15/2022 |

FIPS 140-3 changes

- A lot of deprecated functionality, sometimes in-flight after submitting the module for certification
 - FIPS 186-5 removes DSA completely, published in February 2023
 - FIPS 180-4 is being revised to remove SHA-1 completely by 2030
- Generally, NIST does not look at the protocol level modernisation
 - Certification applies to a vendor-provided crypto modules
 - Compliance is a matter between a vendor, a customer, and a FIPS auditor



**From NIST SHA-1 transition
announcement**

Plan

Before December 31, 2030, NIST plans to:

- Publish FIPS 180-5 (a revision of FIPS 180) to remove the SHA-1 specification,
- Revise [SP 800-131A](#) and other affected NIST publications to reflect the planned withdrawal of SHA-1, and
- Create and publish a transition strategy for the Cryptographic Module Validation Program (CMVP) and the Cryptographic Algorithm Validation Program (CAVP).

Throughout this process, NIST will actively engage with government agencies, validation testing laboratories, vendors, Standards Developing Organizations, sector/industry organizations, users, and other stakeholders to minimize potential impacts and facilitate a smooth transition.

NIST encourages these entities to begin planning for this transition now.


By completing their transition before December 31, 2030, stakeholders – particularly cryptographic module vendors – can help minimize potential delays in the validation process.

Laboratories are opinionated

- Accredited laboratories differ in opinion on NIST guidance
 - SHA-1 not allowed anymore at all *now*
 - Crypto modules cannot instantiate non-well-known curves at all
 - Certain APIs might be asked to be removed by one lab but not the other
 - Certification takes long time, labs anticipate a future guidance change

FIPS 140-3 compliant application cannot interoperate with Active Directory

- Active Directory only supports AES ciphers from RFC 3962
- FIPS 140-3 does not allow
 - Use of RFC 3962 ciphers
 - Use of SHA-1 hashes other than verifying legacy signatures



How these requirements are enforced?

System-wide crypto policy

- `crypto-policies(7)`
- set of rules to derive a crypto module and application configuration
 - allows applying policy specific to a mode OS runs in
 - `DEFAULT`, `FIPS`, `FUTURE`, `LEGACY`, etc.



System-wide crypto policy

- `crypto-policies(7)`
 - Sample generated configurations for [RHEL 8](#), [RHEL 9](#), [Fedora](#)


rhel8 ▾


fedora-crypto-policies / tests / outputs /


+ ▾


Name


..


 DEFAULT-bind.txt


 DEFAULT-gnutls.txt


 DEFAULT-java.txt


 DEFAULT-krb5.txt


 DEFAULT-libreswan.txt


 DEFAULT-libssh.txt

 DEFAULT-nss.txt

 DEFAULT-openssh.txt

 DEFAULT-opensshserver.txt

 DEFAULT-openssl.txt

 DEFAULT-openssl.cnf.txt

System-wide crypto policy

- `crypto-policies(7)`
 - supports sub-policies to tweak the main policy
 - e.g. `AD-SUPPORT`, `NO-SHA1`, ...
 - sub-policies can be combined when defining the system policy
 - e.g. `FIPS:AD-SUPPORT`,
`DEFAULT:NO-SHA1`



System-wide crypto policy

- RHEL 9 DEFAULT policy config for MIT Kerberos

```
[libdefaults]
permitted_enctypes = aes256-cts-hmac-sha1-96 aes256-cts-hmac-sha384-192 aes128-cts-hmac-sha256-128 aes128-cts-hmac-sha1-96
```

- RHEL 9 FIPS 140-3 policy config for MIT Kerberos

```
[libdefaults]
permitted_enctypes = aes256-cts-hmac-sha384-192 aes128-cts-hmac-sha256-128
```




Application-level policy compliance

Application-level compliance

- Crypto modules patched to load a system-wide crypto policy
 - no need to manually define anything in crypto library configuration per each application



Application-level compliance

- Applications would fail when calling non-compliant crypto primitives
 - Applications need to be modernized to stay compliant



Application-level compliance

- Application modernization
 - Defaults
 - Crypto algorithm agility
 - Data migration





Application modernization

Application modernization: defaults

- New installs are easy
 - FreeIPA changed to use `aes256-sha2` as Kerberos master key
 - With the same encryption types as before, default crypto-policy prevents use of banned keys
 - Password change obeys default crypto-policy



Algorithm agility

- Interoperability is hard
 - Protocols aren't updated magically
 - RFC update process might take *years*
 - Adjusting implementations takes *years*
 - Old deployments need to talk to new systems
 - New deployments need to accept old systems



Algorithm agility: PKINIT case

- PKINIT Algorithm Agility, [RFC 8636](#)
 - Adds SHA-1, SHA-256, SHA-512
- PKINIT RFCs requirements
 - DH exchange with ANS X9.42 encoding
 - Certain MODP groups must be supported [RFC 4556](#)
 - [X] [Group 2](#)
 - [X] [Group 14](#)
 - [] [Group 16](#)

Algorithm agility: PKINIT case

- PKINIT RFCs ambiguity
 - digestAlgorithm and signatureAlgorithm might be different
 - Heimdal thinks so, [OpenSSL CMS does not](#)



Algorithm agility: PKINIT case

- PKINIT implementations
 - No full agility in MIT Kerberos yet
 - No ECC support in MIT Kerberos
 - Active Directory does not support new MODP groups
 - Active Directory uses default digest of **SHA-1** when not using ECC
 - Heimdal defaults to MODP group 2



Algorithm agility: PKINIT case

- FIPS 140-3 enforcement with MIT Kerberos
 - Relies on OpenSSL implementation
 - Two OpenSSL FIPS providers (upstream and RHEL downstream)
 - Different labs to certify both crypto modules
 - What can go wrong?



Algorithm agility: PKINIT case

- FIPS 140-3 enforcement with MIT Kerberos
 - Heimdal defaults to MODP group 2
 - OpenSSL cannot decrypt this group
 - Heimdal provides MODP group 14 but OpenSSL fails earlier
 - OpenSSL breaks interoperability



Algorithm agility: PKINIT case

- FIPS 140-3 enforcement with MIT Kerberos
 - Disable **SHA-1**, no way to verify Windows PKINIT clients
 - Same to older RHEL 7/8, defaults to **SHA-1**



Algorithm agility: PKINIT case

- FIPS 140-3 enforcement with MIT Kerberos
 - Move default to `SHA-256` for `supportedCMSTypes`
 - no way to verify Windows, old RHEL 7/8



Algorithm agility: PKINIT case

- FIPS 140-3 enforcement with MIT Kerberos
 - Switch dynamically between OpenSSL crypto providers depending on the client
 - allows to support legacy clients if system-wide crypto policy permits
 - `FIPS:AD-SUPPORT-LEGACY`



Algorithm modernization: Active Directory

- Move to newer AES-based ciphers
 - new variations of SAMR, LSA, NETLOGON calls
- Tighten up use of crypto material
 - `samba.trust_utils.CreateTrustedDomainRelax`
 - used by FreeIPA and `samba-tool`
- Still not enough for FIPS 140-3

The background of the image is a close-up, top-down view of a wooden shingle roof. The shingles are made of light-colored wood and are arranged in a traditional overlapping pattern. A semi-transparent blue circle is centered on the image, serving as a backdrop for the text.

Data migration

Samba AD data migration

- **Imaginary case for future Samba AD**
- Access to old encrypted keys requires extended crypto policy
 - e.g. `FIPS:AD-SUPPORT-LEGACY` before migration, then `FIPS:AD-SUPPORT`
- Active Directory DC case:
 - Kerberos keys need to be regenerated
 - Plain-text passwords exist, offline regeneration possible?
- Active Directory domain member case:
 - Machine account password / keytab regeneration

FreeIPA data migration

- Access to old encrypted keys requires extended crypto policy
 - e.g. `FIPS:AD-SUPPORT-LEGACY` before migration, then `FIPS:AD-SUPPORT`
- FreeIPA DC case:
 - No plain-text passwords exist, full password/key refresh is needed
 - time to move to passwordless?
 - Keytabs with service keys need to be rotated, can be automated

FreeIPA data migration

- FreeIPA client case:
 - Keytabs with host keys need to be rotated, can be automated
 - Passwordless service update using certificates:
 - Map certificate to a service
 - use PKINIT authentication to obtain a Kerberos ticket
 - rotate Kerberos service keys



Questions?

Mastodon: @abbra:mastodon.social

Blog: vda.li/en