3.13.11

FIPS 140-2

Basic Information

Just because the OS (Linux, or Windows) implements and supports FIPS compliance it does not guarantee that every application installed is also compliant.

WINDOWS

Windows implements FIPS compliance by turning on the GPO “System cryptography: Use FIPS compliant algorithms for encryption, hashing, and signing”.

This only pertains to the operating system. Any third party application can implement their own cryptography, although Microsoft recommends vendors point their applications to using the certified modules in the Windows OS.

LINUX

The FIPS 140-2 standard ensures that cryptographic tools implement their algorithms properly. See the full FIPS 140-2 standard at http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf for further details on these levels and the other specifications of the FIPS standard.

To fulfil the strict FIPS 140-2 compliance, add the fips=1 kernel option to the kernel command line during system installation. With this option, all keys' generations are done with FIPS-approved algorithms and continuous monitoring tests in place. After the installation, the system is configured to boot into FIPS mode automatically.

After the System Installation

To turn your system, kernel and user space, into FIPS mode anytime after the system installation, follow these steps:

Make sure prelinking is disabled.

For proper operation of the in-module integrity verification, prelinking of libraries and binaries has to be disabled. Prelinking is done by the prelink package, which is not installed by default. To disable prelinking, set the PRELINKING=no option in the /etc/sysconfig/prelink configuration file. To disable existing prelinking on all system files, use the prelink -u -a command.

Install the dracut-fips package:

~]# yum install dracut-fips

For the CPUs with the AES New Instructions (AES-NI) support, install the dracut-fips-aesni package as well:

~]# yum install dracut-fips-aesni

Regenerate the initramfs file.

To enable the in-module integrity verification and to have all required modules present during the kernel boot, the initramfs file has to be regenerated:

~]# dracut -v -f

Warning

This operation will overwrite the existing initramfs file.

Modify boot loader configuration.

To boot into FIPS mode, add the fips=1 option to the kernel command line of the boot loader. If your /boot or /boot/EFI/ partitions reside on separate partitions, add the boot=<partition> (where <partition> stands for /boot or /boot/EFI) parameter to the kernel command line as well.

To identify the boot partition, enter the following command:

~]$ df /boot

Filesystem 1K-blocks Used Available Use% Mounted on

/dev/sda1 495844 53780 416464 12% /boot

To ensure that the boot= configuration option works even if the device naming changes between boots, identify the universally unique identifier (UUID) of the partition by running the following command:

~]$ blkid /dev/sda1

/dev/sda1: UUID="05c000f1-f899-467b-a4d9-d5ca4424c797" TYPE="ext4"

Append the UUID to the kernel command line:

boot=UUID=05c000f1-f899-467b-a4d9-d5ca4424c797

Depending on your boot loader, make the following changes:

grub2

Add the fips=1 and boot=<partition of /boot or /boot/EFI> options to the GRUB\_CMDLINE\_LINUX key in the /etc/default/grub file. To apply the changes to /etc/default/grub, rebuild the grub.cfg file as follows:

On BIOS-based machines, enter the following command as root:

~]# grub2-mkconfig -o /boot/grub2/grub.cfg

On UEFI-based machines, enter the following command as root:

~]# grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg

zipl (on the IBM z Systems architecture only)

Add the fips=1 and boot=<partition of /boot> options to the /etc/zipl.conf to the kernel command line and apply the changes by running the following command as root:

~]# zipl

Reboot your system.