Moxa Industrial Linux 3.0 (Debian 11) Manual for ioThinx 4530 Series

Version 1.0, April 2024

www.moxa.com/products

Moxa Industrial Linux 3.0 (Debian 11) Manual for ioThinx 4530 Series

Copyright Notice

Trademarks

Disclaimer

- Information in this document is subject to change without notice and does not represent a commitment on the part of Moxa.
- Moxa provides this document as is, without warranty of any kind, either expressed or implied, including, but not limited to, its particular purpose. Moxa reserves the right to make improvements and/or changes to this manual, or to the products and/or the programs described in this manual, at any time.
- Information provided in this manual is intended to be accurate and reliable. However, Moxa assumes no responsibility for its use, or for any infringements on the rights of third parties that may result from its use.
- This product might include unintentional technical or typographical errors. Changes are periodically made to the information herein to correct such errors, and these changes are incorporated into new editions of the publication.

Technical Support Contact Information

www.moxa.com/support

Table of Contents

1. Introduction

Moxa Industrial Linux 3

Moxa Industrial Linux 3 (MIL3) is an industrial-grade Linux distribution developed and maintained by Moxa to address the security, reliability, and long-term support needs of industrial automation systems such as transportation, energy, oil and gas, and manufacturing.

MIL3 is based on Debian 11 with kernel 5.10 and integrated with several feature sets designed to strengthen and accelerate user application development as well as ensure system reliability and security.

2. Getting Started

Connecting to the Arm-based Computer

You will need another computer to connect to the Arm-based computer and log on to the command-line interface. There are two ways to connect: locally through serial console or ethernet cable, or remotely via Secure Shell (SSH). Refer to the Hardware Manual to see how to set up the physical connections.

For default login username and password, please reference the Default Credentials and Password Strength.

The username and password are the same for all serial console and SSH remote log in actions. Root account login is disabled until you manually create a password for the account. The user **moxa** is in the **sudo** group so you can operate system level commands with this user using the **sudo** command. For additional details, see the Sudo Mechanism section in Chapter 7.

ATTENTION

For security reasons, we highly recommend that you disable the default user account and create your own user accounts.

Connecting through the Serial Console

This method is particularly useful when using the computer for the first time. The signal is transmitted over a direct serial connection, so you do not need to know either of its two IP addresses to connect to the Arm-based computer. To connect through the serial console, configure your PC's terminal software using the following settings.

Serial Console Port Settings	
Baudrate	115200 bps
Parity	None
Data bits	8
Stop bits	1
Flow Control	None
Terminal	VT100

Below we show how to use the terminal software to connect to the Arm-based computer in a Linux environment and in a Windows environment.

Linux Users

> **NOTE**
>
> These steps apply to the Linux PC you are using to connect to the Arm-based computer. Do NOT apply these steps to the Arm-based computer itself.

Take the following steps to connect to the Arm-based computer from your Linux PC.

1. Install **minicom** from the package repository of your operating system.

   For Centos and Fedora:

   ```
   user@PC1:~# yum -y install minicom
   ```

   For Ubuntu and Debian:

   ```
   user@PC2:~# apt install minicom
   ```

2. Use the **`minicom -s`** command to enter the configuration menu and set up the serial port settings.

   ```
   user@PC1:~# minicom -s
   ```

3. Select **Serial port setup**.

   ```
   +-----[configuration]------+
   | Filenames and paths      |
   | File transfer protocols  |
   | Serial port setup        |
   | Modem and dialing        |
   | Screen and keyboard      |
   | Save setup as dfl        |
   | Save setup as..          |
   | Exit                     |
   | Exit from Minicom        |
   +--------------------------+
   ```

4. Select **A** to change the serial device. Note that you need to know which device node is connected to the Arm-based computer.

   ```
   +-----------------------------------------------------------------------+
   | A -    Serial Device      : /dev/tty8                                 |
   | B - Lockfile Location     : /var/lock                                 |
   | C -   Callin Program      :                                           |
   | D -  Callout Program      :                                           |
   | E -    Bps/Par/Bits       : 115200 8N1                                |
   | F - Hardware Flow Control : No                                        |
   | G - Software Flow Control : No                                        |
   |                                                                       |
   |    Change which setting?                                              |
   +-----------------------------------------------------------------------+
           | Screen and keyboard      |
           | Save setup as dfl        |
           | Save setup as..          |
           | Exit                     |
           | Exit from Minicom        |
           +--------------------------+
   ```

5. Select **E** to configure the port settings according to the **Serial Console Port Settings** table provided.
6. Select **Save setup as dfl** (from the main configuration menu) to use default values.
7. Select **Exit from minicom** (from the configuration menu) to leave the configuration menu.
8. Execute **minicom** after completing the above configurations.

   ```
   user@PC1:~# minicom
   ```

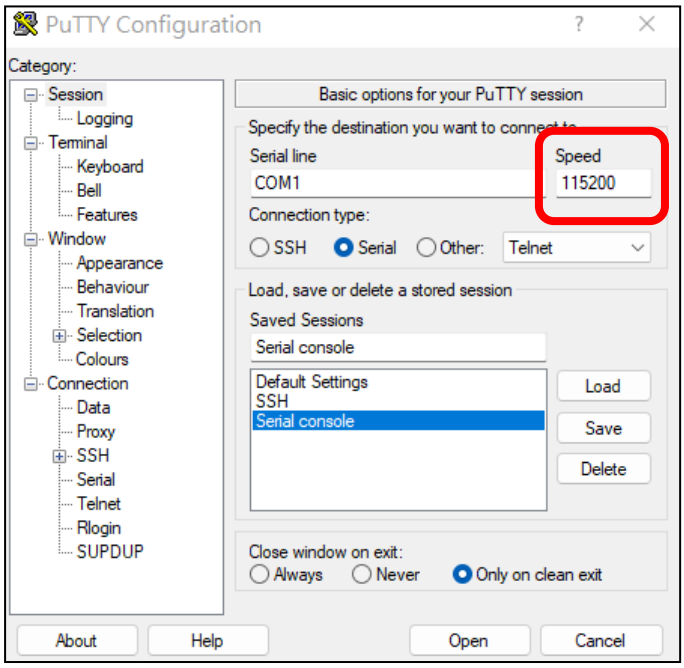
Windows Users

NOTE

These steps apply to the Windows PC you are using to connect to the Arm-based computer. Do NOT apply these steps to the Arm-based computer itself.

Take the following steps to connect to the Arm-based computer from your Windows PC.

1. Download PuTTY http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html to set up a serial connection with the Arm-based computer in a Windows environment. The figure below shows a simple example of the configuration that is required.
2. Once the connection is established, the following window will open.



3. Select the **Serial** connection type and choose settings that are similar to the Minicom settings.

Connecting via the SSH

The Arm-based computer supports SSH connections remotely or over an Ethernet network. If you are connecting the computer using an Ethernet cable, refer to the following IP addresses information.

Ethernet Port	Configuration	IP Address
LAN 1	Static IP	192.168.3.127
LAN 2	Static IP	192.168.4.127

NOTE

Be sure to configure the IP address of your notebook/PC's Ethernet interface on the same subnet as the LAN port of Arm-based computer you plan to connect to. For example, 192.168.4.**126** for LAN2.

Linux Users

NOTE

These steps apply to the Linux PC you are using to connect to the Arm-based computer. Do NOT apply these steps to the Arm-based computer itself.

Use the **ssh** command from a Linux computer to access the computer's LAN2 port.

```
user@PC1:~ ssh moxa@192.168.4.127
```

Type **yes** to complete the connection.

```
The authenticity of host '192.168.4.127' can't be established.
RSA key fingerprint is 8b:ee:ff:84:41:25:fc:cd:2a:f2:92:8f:cb:1f:6b:2f.
Are you sure you want to continue connection (yes/no)? yes_
```

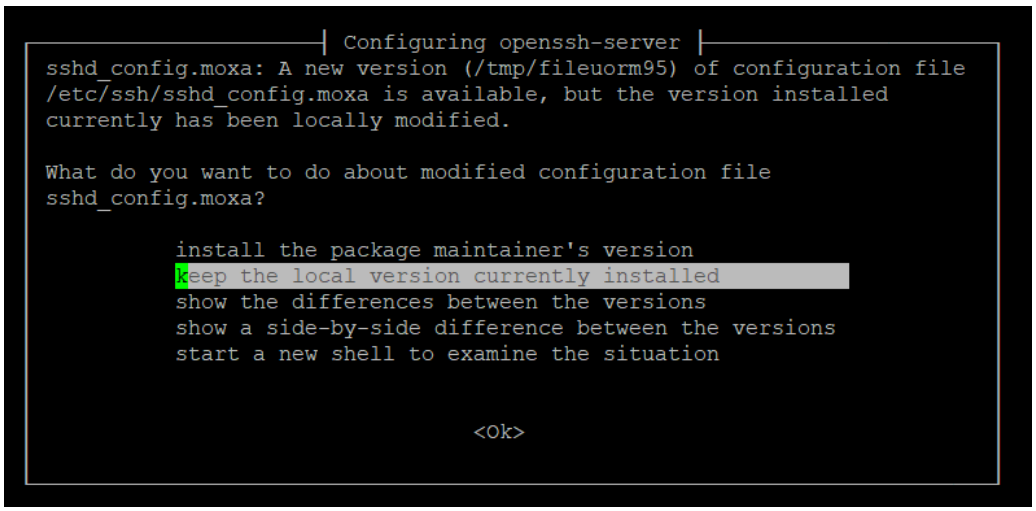
ATTENTION

Regenerate SSH key regularly

In order to secure your system, we suggest doing a regular SSH-rekey, as shown in the following steps:

```
moxa@moxa-tbzkb1090923:~$ cd /etc/ssh
moxa@moxa-tbzkb1090923:~$ sudo rm /etc/ssh/ssh_host_*
moxa@moxa-tbzkb1090923:~$ sudo dpkg-reconfigure openssh-server
moxa@moxa-tbzkb1090923:~$ sudo systemctl restart ssh
```

Select "**keep the local version currently installed**" following is prompt during rekey process



For more information about SSH, refer to the following link.

https://wiki.debian.org/SSH

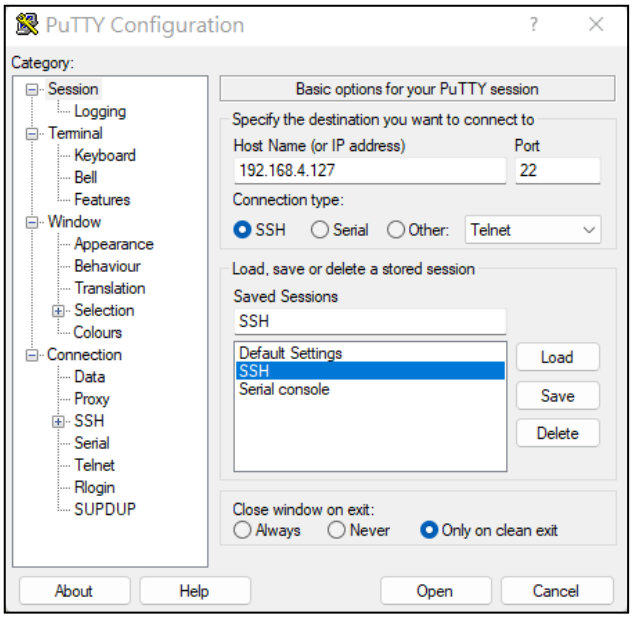
Windows Users

NOTE

These steps apply to the Windows PC you are using to connect to the Arm-based computer. Do NOT apply these steps to the Arm-based computer itself.

Take the following steps from your Windows PC.

Click on the link http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html to download PuTTY (free software) to set up an SSH console for the Arm-based computer in a Windows environment. The following figure shows a simple example of the configuration that is required.



Managing User Accounts

Default User Account and Password Policy

The default login username and password of Moxa Industrial Linux are both **moxa** for the first-time login. You will be prompted to set a new password before you can continue to login.

- Default Username: **moxa**
- Default Password: **moxa**

Password Strength Requirements:

- At least 8 characters in length
- Dictionary checking is enabled to prevent the use of common passwords

To modify the password strength policy, edit the **/etc/security/pwquality.conf.d/00-moxa-standard-pwquality.conf** file to configure the policy.

NOTE

Click the following link for more information on the password strength configuration.

https://manpages.debian.org/bullseye/libpwquality-common/pwquality.conf.5.en.html

For bootloader administrator password configuration, refers to the bootloader configuration section.

Creating and Deleting User Accounts

ATTENTION

DO NOT disable the default account before creating an alternative user account.

You can use the **useradd** and **userdel** commands to create and delete user accounts. Be sure to reference the main page of these commands to set relevant access privileges for the account. Following example shows how to create a **test1** user in the **sudo** group whose default login shell is **bash** and has home directory at **/home/test1**:

```
moxa@ moxa-tbzkb1090923:~# sudo useradd -m -G sudo -s /bin/bash test1
```

To change the password for test1, use the **passwd** option along with the new password. Retype the password to confirm the change.

```
moxa@moxa-tbzkb1090923:~# sudo passwd test1
New password:
Retype new password:
passwd: password updated successfully
```

To delete the user **test1**, use the **userdel** command.

```
moxa@ moxa-tbzkb1090923:# sudo userdel test1
```

Modifying User Accounts

You can use the **usermod** commands to create and modify the user account settings. Some examples of commonly used settings are listed here, including adding a user to a group, locking an account, activating an account and setting the password expiration date for the account.

1. Adding user test1 to the user group Moxa

   ```
   moxa@ moxa-tbzkb1090923:# sudo usermod -a -G Moxa test1
   ```

2. Disabling or locking the user account test1

   ```
   moxa@ moxa-tbzkb1090923:# sudo usermod -L test1
   ```

3. Activating the user account test1

   ```
   moxa@ moxa-tbzkb1090923:# sudo usermod -U test1
   ```

4. Set a password expire date of 2023-11-01 for the user account test1.

   ```
   moxa@ moxa-tbzkb1090923:# sudo usermod -e 2023-11-01 test1
   ```

NOTE

Refers to below link for complete usage of **usermod**

https://linux.die.net/man/8/usermod

Changing the Password

You can use the **passwd** commands to change the password of a user account. Changing the password will not have any impact on other functionalities.

An example of changing the password for user account **test1**.

```
moxa@ moxa-tbzkb1090923:# sudo passwd test1
New password:
Retype new password:
passwd: password updated successfully
```

Querying the System Image Version

Use the **mx-ver** command to check the system **image version** on your Arm-based computer.

```
moxa@moxa-tbzkb1090923:# mx-ver
ioThinx 4533-LX MIL3 version 2.0.0 Build 24011609
```

```
moxa@moxa-tbzkb1090923:# mx-ver -h

Usage: mx-ver [OPTION]
       -a: show product information inline
       -b: show the build time
       -m: show the model name
       -v: show the image version
       -A: show all information
       -M: show the MIL version
       -o: show the image option code
       -h: show the help menu
```

Querying the Device Information

Use the **# mx-interface-mgmt deviceinfo** command to retrieve general information for your Moxa Arm-based Computer.

Command and Usage	Description
deviceinfo	Shows the following device information: • Serial number (S/N) • Model name. • SECUREBOOT (Enabled/Disabled)

(* Secure boot is not supported in ioThinx 4533)

```
moxa@moxa-tbbbb1182827:~$ mx-interface-mgmt deviceinfo
SERIALNUMBER=TBCKB1045627
MODELNAME=ioThinx 4533-LX
SECUREBOOT=Disabled
```

Determining Available Drive Space

To determine the amount of available drive space, use the **df** command with the **–h** option. The system will return the amount of drive space broken down by file system. Here is an example:

```
moxa@moxa-tbzkb1090923:~$ sudo df -h

Filesystem      Size  Used Avail Use% Mounted on
udev            979M     0  979M   0% /dev
tmpfs          1005M  932K 1004M   1% /run
/dev/mmcblk2p2  982M  270M  657M  30% /boot_device/p2
/dev/mmcblk2p3  5.8G   61M  5.4G   2% /boot_device/p3
/dev/mmcblk2p4  238M   32M  190M  15% /var/log
/dev/loop0      270M  270M     0 100% /boot_device/p2/lower
overlay         5.8G   61M  5.4G   2% /
/dev/mmcblk2p1  115M   30M   77M  28% /boot_device/p1
tmpfs          1005M     0 1005M   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           201M     0  201M   0% /run/user/1000
```

Shutting Down the Device

To shut down the computer, first disconnect the power source. When the computer is powered off, main components such as the CPU, RAM, and storage devices are powered off, although an internal clock may retain battery power.

You can use the Linux command **shutdown** to close all software running on the device and halt the system. However, main components such as the CPU, RAM, and storage devices will continue to be powered after you run this command.

```
moxa@moxa-tbzkb1090923: ~# sudo shutdown -h now
```

3. Device Configuration

In this chapter, we describe how to configure the basic settings of Moxa Arm-based computers, including using the bootloader menu, configuring the network connections and power-saving settings, and localizing the computer. The instructions in this chapter cover all functions supported in Moxa Arm-based computers. Before referring to the sections in this chapter, ensure that they are applicable to and are supported by the hardware specification of your Arm-based computer.

Bootloader Configuration

Accessing the Bootloader Configuration Menu

To access bootloader menu, you must first connect to Moxa Arm-based computer via its serial console port. After powering on the Arm-based computer, press **Ctrl + Backspace** or **DEL** to enter the bootloader configuration menu.

NOTE

If you cannot enter the bootloader menu by pressing <DEL>, replace the PuTTy tool with the Tera Term terminal console tool (detailed information is available at: https://ttssh2.osdn.jp/index.html.en.)

```
--------------------------------------------------------------------------------
 Model: ioThinx 4533-LX
 Boot Loader Version: 2.1.0S00
 Build date: Nov 24 2023 - 13:59:32    Serial Number: TBCKB1045627
 LAN1 MAC: 00:90:E8:C0:3F:58           LAN2 MAC: 00:90:E8:C0:3F:59
--------------------------------------------------------------------------------
 (0) Boot Management                   (1) Install System Image
 (2) Admin Password                    (3) Advance Setting
 (4) Exit & Reboot                     (5) Go To Linux
--------------------------------------------------------------------------------
```

Boot Management

Boot Option

By default, Moxa Arm-based computers boot up from the embedded eMMC flash, it also provides an option to boot up from an external SD.

The following is an example of changing the first boot priority to SD card and setting the secondary boot option to SD card if the first option fails to boot.

1. Select **(0) Boot Management > (1) Boot Option**
2. Choose to first boot from an external storage.
3. Choose if the embedded storage should be disabled.

   If the embedded storage is disabled, Moxa Arm-based computers will only attempt to boot from the SD card. If embedded storage is set to eMMC, the computers will try to boot from SD; if that fails, they will boot from eMMC.

4. Set the External Storage to the SD card.

```
--------------------------------------------------------------------------------
 Model: ioThinx 4533-LX
 Boot Loader Version: 2.1.0S00
 Build date: Nov 24 2023 - 13:59:32    Serial Number: TBCKB1045627
 LAN1 MAC: 00:90:E8:C0:3F:58           LAN2 MAC: 00:90:E8:C0:3F:59
--------------------------------------------------------------------------------
 (0) Boot Management                   (1) Install System Image
 (2) Admin Password                    (3) Advance Setting
 (4) Exit & Reboot                     (5) Go To Linux
--------------------------------------------------------------------------------

Command>>1
Boot Management : Boot Option
Boot Order : Embedded First
Embedded Storage : eMMC
External Storage : SD

Would you like to configure the Boot Option?

0 - No, 1 - Yes (0-1, Enter to abort): 1

Set Boot Order:
 0 - Embedded First, 1 - External First (0-1, Enter to abort): 1

Set Embedded Storage:
 0 - Disabled, 1 - eMMC (0-1, Enter to abort): 1

Set External Storage:
 0 - Disabled, 1 - SD, 2 - USB (0-2, Enter to abort): 1
Writing to MMC(2)...
Boot Management : Boot Option
Boot Order : External First
Embedded Storage : eMMC
External Storage : SD
```

The table below lists all possible combinations of boot options configuration and the corresponding boot action.

Set Boot Order	Set Embedded Storage	Set External Storage	Boot Action
0 – Embedded First	1 – eMMC	0 – Disabled	Boot from eMMC
1 – External First	0 – Disabled	1 – SD	Boot from the external storage
0 – Embedded First	1 – eMMC	1 – SD	First boot from eMMC; if it fails, boot from the external storage
1 – External First	1 – eMMC	1 – SD	Boot from the external storage; if this fails, boot from eMMC

Advance Boot Option

Allow advanced users to edit the **bootargs** and **bootcmd** parameters to customize the boot process.

- **bootargs:** Used to tell the kernel how to configure various device drivers and where to find the root filesystem.
- **bootcmd:** Bootloader will execute the commands listed sequentially. Commands should be separated by semicolons.

Installing the System Image

Installing System Image From TFTP

1. Prepare a TFTP server.
2. Set up a TFTP server.
3. Make sure the image (*.img) and hash file (*.img.sha256sum.bin) is in your TFTP server directory.

IMPORTANT!

Use this method to install a system image on your computer if the size of the image file is less than 2 GB. If the file size is larger than 2 GB, use the SD card to install the system image.

4. In bootloader menu, select (1) **Install System Image > (3) TFTP Settings** and configure the following:
   - The LAN port to be used for TFTP transfer.
   - Local IP address of LAN port.
   - TFTP server IP.

```
--------------------------------------------------------------------------------
 Model: ioThinx 4533-LX
 Boot Loader Version: 2.1.0S00
 Build date: Nov 24 2023 - 13:59:32    Serial Number: TBCKB1045627
 LAN1 MAC: 00:90:E8:C0:3F:58           LAN2 MAC: 00:90:E8:C0:3F:59
--------------------------------------------------------------------------------
 (0) Install System Image from TFTP    (1) Install System Image from SD
 (2) Install System Image from USB     (3) TFTP Settings
--------------------------------------------------------------------------------
```

5. Press **ESC** to exit and select **Install System Image from TFTP**.

```
If you want to change the TFTP IP address, enter 1 to set up the local LAN
port IP address and the TFTP server IP address, and then choose an image
(*.img) file.Current IP Address

Local IP Address : 192.168.5.132
Server IP Address : 192.168.5.133
Using LAN2 to download data.
Do you want to change the ip address?
0 - No, 1 - Yes (0-1, Enter to abort): 1
Local IP Address : 192.168.4.127
Server IP Address : 192.168.4.100
Writing to MMC(2)...
System Image File Name (system image.img): IMG_ioThinx-
4530_MIL3IOTHINX_2.0.0_Build_24011609.img
```

6. After the system image installation process is complete, unplug the power supply and reboot the system.
7. After rebooting the system, you can use the following command to check if the system image is up to date.

```
moxa@moxa-tbckb1045623:~$ mx-ver
ioThinx 4533-LX MIL3 version 2.0.0 Build 24011609
```

Installing the System Image From SD

The system image on the ioThinx 4533 can be installed through an external SD. Prepare an SD disk in the FAT32 or ext4 format with the system image and plug it into the SD port of the computer.

1. Select **Install System Image > Install System Image from SD.**
2. Type in the system image file name.

```
--------------------------------------------------------------------------------
 Model: ioThinx 4533-LX
 Boot Loader Version: 2.1.0S00
 Build date: Nov 24 2023 - 13:59:32    Serial Number: TBCKB1045623
 LAN1 MAC: 00:90:E8:C0:3F:50           LAN2 MAC: 00:90:E8:C0:3F:51
--------------------------------------------------------------------------------
 (0) Install System Image from TFTP    (1) Install System Image from SD
 (2) Install System Image from USB     (3) TFTP Settings
--------------------------------------------------------------------------------
Command>>1

System image File Name (System_image.img): IMG_ioThinx-
4530_MIL3IOTHINX_2.0.0_Build_24011609.img
```

> **NOTE**
>
> Make sure to put **the hash file of the system image** in the same folder as the image as integrity validation is required.

3. After the system image installation process is complete, unplug the power supply and reboot the system.
4. After rebooting the system, you can use the following command to check if the system image is up to date.

```
moxa@moxa-tbckb1045623:~$ mx-ver
ioThinx 4533-LX MIL3 version 2.0.0 Build 24011609
```

Administrator Password

Enabling/Disabling Admin Password

The bootloader menu is not protected by password by default. To enhance the security of your Moxa Arm-based computer, it is strongly recommended to set up an administrator password if there is physical unauthorized access is possible. To set up an administrator password, follow the below procedures:

1. Select **Admin Password > Enable/Disable Admin Password**.
2. Select **1** to set up an administrator password. If **0** (disable) is selected, the currently set password will be cleared.
3. Enter the password you would like to set twice; the password strength requirement is at least 8 characters in length.

```
--------------------------------------------------------------------------------
 Model: ioThinx 4533-LX
 Boot Loader Version: 2.1.0S00
 Build date: Nov 24 2023 - 13:59:32    Serial Number: TBCKB1045627
 LAN1 MAC: 00:90:E8:C0:3F:58           LAN2 MAC: 00:90:E8:C0:3F:59
--------------------------------------------------------------------------------
 (0) Set to Default                    (1) Enable/Disable Admin Password
 (2) Configure Admin Password          (3) Configure Admin Password Policy
--------------------------------------------------------------------------------
Command>>1
Current Mode: Disabled

0 - Disable, 1 - Enable (0-1, Enter to abort): 1
```

```
The current password is empty, please set one.

Admin Password Policy:
 - Minimum length: 8

Enter new password: **********
Retype password: **********
Password set successfully

Password status : Enabled.
```

4. Once Administrator password is set, password authentication is required when accessing bootloader menu.

```
Press <DEL> To Enter BIOS configuration Setting

Enter the Administrator password
Enter current password: **********
```

WARNING

It is important to save the password in a secure location. If the password is lost and access to bootloader menu is needed, you will have to contact Moxa technical support to send your Arm-based computer to Moxa for password reset.

Configuring the Admin Password Policy

To change the administrator password, select **Admin Password > Configure Admin Password** and follow the on-screen instructions. Changing the password will not have any impact on functionalities.

```
------------------------------------------------------------------------------
 Model: ioThinx 4533-LX
 Boot Loader Version: 2.1.0S00
 Build date: Nov 24 2023 - 13:59:32    Serial Number: TBCKB1045627
 LAN1 MAC: 00:90:E8:C0:3F:58           LAN2 MAC: 00:90:E8:C0:3F:59
------------------------------------------------------------------------------
 (0) Set to Default                    (1) Enable/Disable Admin Password
 (2) Configure Admin Password          (3) Configure Admin Password Policy
------------------------------------------------------------------------------
Command>>3

Current setting:
Admin Password Policy:
 - Minimum length: 8


*******************************************************************************
*

 Do you want to configure admin password policy setting?

*******************************************************************************
*
0 - No, 1 - Yes (0-1, Enter to abort): 1
- Minimum length (6-16, Enter to abort): 6
- Minimum numeric numbers (0-16, Enter to abort): 1
- Minimum lowercase or uppercase letters combined (0-16, Enter to abort): 1
```

Minimum Length

Setting	Description	Factory Default
Input from 6 to 16	It allows users to decide the minimum length of the password.	8

Minimum Numeric Numbers

Setting	Description	Factory Default
Input from 0 to 16	It allows users to decide the minimum of numeric number that the password must contain	0

Minimum Lowercase or Uppercase Letters Combined

Setting	Description	Factory Default
Input from 0 to 16	It allows users to decide the minimum letters (lowercase or uppercase combined) that the password must contain.	0

Configuring Admin Password

To change the administrator password, select **Admin Password > Configure Admin Password** and follow the on-screen instructions.

Resetting the Admin Password to Default

If you lost your password, follow the below steps to reset the password to the factory default:

1. After powering on the Arm-based computer, press **Ctrl + Backspace** or **DEL** to enter the Bootloader configuration menu that prompts for a password.

```
Press <DEL> To Enter BIOS configuration Setting

Enter the Administrator password
Enter current password: **********
```

2. Immediately press and hold the **Reset** button on the ioThinx 4533 for over 5 seconds will trigger the password reset process. You must complete this step within **10 seconds** after step one for the reset process to initiate.

Login Policy

Invalid Login Attempts

This determines the **maximum consecutive failure login attempts** allowed during the specified **time period** and the duration to block users from accessing bootloader configuration menu when failure login attempts and time period are over the defined threshold.

To configure this policy, select **Advance Setting > Configure Invalid Login Attempts** and follow the on-screen instructions.

```
--------------------------------------------------------------------------------
 Model: ioThinx 4533-LX
 Boot Loader Version: 2.1.0S00
 Build date: Nov 24 2023 - 13:59:32    Serial Number: TBCKB1045627
 LAN1 MAC: 00:90:E8:C0:3F:58           LAN2 MAC: 00:90:E8:C0:3F:59
--------------------------------------------------------------------------------
 (0) Set to Default                    (1) Configure Auto Reboot
 (2) Configure Login Message           (3) Configure Invalid Login Attempts
 (4) Clear TPM                         (5) Configure Linux Security Modules
 (6) View Bootloader log
--------------------------------------------------------------------------------
Command>>3
Current setting: [5] consecutive invalid login within [60] seconds will reboot
and disable access to bootloader menu for [300] seconds.
********************************************************************************
*
 Do you want to configure the invalid login attempts setting?
********************************************************************************
*
0 - No, 1 - Yes (0-1, Enter to abort): 1

Input 0 to any of the configuration below will disable invalid login check
```

```
Consecutive invalid login attempts (0-5, Enter to abort):
Within how many seconds (0-60, Enter to abort):
Disable access for how many seconds (0-900, Enter to abort):
```

Consecutive Invalid Login Attempts

Configuration	Setting	Factory Default
Consecutive invalid login attempts	Input from 0 to 5	0
Within how many Seconds	Input from 0 to 60	0
Disable access for how many seconds	Input from 0 to 900	0

NOTE

Input 0 to any of the above configurations will disable the invalid login check.

Auto Reboot After Inactivity

This determines the time period for auto reboot when users do not do any action.

To set the time period, select **(2) Advance Setting > (1) Configure Auto Reboot** and follow the on-screen instructions.

Setting	Description	Factory Default
Input from 0 to 900 (seconds)	This determines the time period for auto reboot when users do not do any action	0

Login Banner Message

This allows users to customize the login message before prompting the administrator password.

To configure the message, select **Advance Setting > Configure Login Message** and follow the on-screen instructions.

```
U-Boot 2020.04-ga174fe3ef0-dirty (May 13 2022 - 14:23:01 +0800)
DRAM:  2 GiB
PMIC: PFUZE3000 DEV_ID=0x31 REV_ID=0x11
MMC:   FSL_SDHC: 0, FSL_SDHC: 2
Loading Environment from SPI Flash... SF: Detected mx25l12805d with page size
256 Bytes, erase size 64 KiB, total 16 MiB
OK
In:    serial
Out:   serial
Err:   serial
SEC0: RNG instantiated
Net:   eth0: ethernet@30be0000 [PRIME]Get shared mii bus on ethernet@30bf0000
FEC0:1 is connected to ethernet@30be0000.  Reconnecting to ethernet@30bf0000
, eth1: ethernet@30bf0000
Model: 0x00
Normal Boot

Press <DEL> To Enter BIOS configuration Setting

Enter the Administrator password
Enter current password:
```

Clearing the TPM Module

Clearing the TPM will erase information stored on the TPM. You will lose all created keys and access to data encrypted by these keys.

To clear the TPM, select **Advance Setting > Clear TPM** and follow the directions.

Localizing Your Arm-based Computer

Adjusting the Time

The Arm-based computer has two time settings. One is the system time, and the other is the RTC (Real Time Clock) time kept by the Arm-based computer's hardware. Use the **`date`** command to query the current system time or set a new system time. Use the **`hwclock`** command to query the current RTC time or set a new RTC time.

Use the **`date MMDDhhmmYYYY`** command to set the system time:

MM = Month
DD = Date
hhmm = hour and minute

```
moxa@moxa-tbzkb1090923:# sudo date 102900282021
Fri 29 Oct 2021 12:28:00 AM GMT
```

Use the following command to set the RTC time to system time:

```
moxa@moxa-tbzkb1090923:# sudo hwclock -w
moxa@moxa-tbzkb1090923:# sudo hwclock
2021-10-28 16:25:04.077432+00:00
```

NOTE

Click the following links for more information on date and time:

https://www.debian.org/doc/manuals/system-administrator/ch-sysadmin-time.html

https://wiki.debian.org/DateTime

NTP Time Synchronization

The Moxa Industrial Linux (MIL) uses Network Time Security (NTS) to secure NTP, which provides a handshake (TLS) before using a NTP server and authentication of the NTP time synchronization packets using the results of the TLS handshake.

The default NTP client in MIL is **`Chrony`**. MIL disabled NTP server without NTS support by default and uses the following public NTP servers that support NTS.

- Cloudflare
- Netnod
- System76
- PTB

The default server list is configured in the **/etc/chrony/sources.d/moxa-nts.sources** file.

```
# prefer nts over ntp server
server time.cloudflare.com nts iburst prefer
server sth1.nts.netnod.se nts iburst prefer
server sth2.nts.netnod.se nts iburst prefer
server virginia.time.system76.com nts iburst prefer
server ohio.time.system76.com nts iburst prefer
server oregon.time.system76.com nts iburst prefer
server ptbtime1.ptb.de nts iburst prefer
server ptbtime2.ptb.de nts iburst prefer
server ptbtime3.ptb.de nts iburst prefer
```

The configuration file for Chrony is at **/etc/chrony/chrony.conf**.

The following example shows some basic functions to monitor the current status of the Chrony's chronyc tool and make changes if necessary.

1. Check the time synchronization status between the local system and the reference server using the command:

   # chronyc tracking

```
moxa@moxa-tbbbb1182827:~$ chronyc tracking
Reference ID    : A29FC801 (time.cloudflare.com)
Stratum         : 4
Ref time (UTC)  : Sun Jul 31 18:27:42 2022
System time     : 0.000334575 seconds slow of NTP time
Last offset     : +0.000226902 seconds
RMS offset      : 0.005672113 seconds
Frequency       : 27.766 ppm fast
Residual freq   : -0.065 ppm
Skew            : 3.403 ppm
Root delay      : 0.203054637 seconds
Root dispersion : 0.006750254 seconds
Update interval : 517.4 seconds
Leap status     : Normal
```

2. Check the time source configured in the **/etc/chrony/chrony.conf** file using the **# chronyc sources** command.

```
moxa@moxa-tbbbb1182827:~$ chronyc sources

MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
^+ ohio.time.system76.com      2   9   377   147    +18ms[  +18ms] +/-  141ms
^+ oregon.time.system76.com    2   9   377   203    +14ms[  +14ms] +/-  137ms
^- ptbtime1.ptb.de             1   9    21   682  -2780us[-2417us] +/-  166ms
^- ptbtime2.ptb.de             1   9    21   674  -5243us[-4882us] +/-  169ms
^- ptbtime3.ptb.de             1   9    21   687    +17ms[  +17ms] +/-  192ms
^+ sth1-ts.nts.netnod.se       1   9   377   220    -12ms[  -12ms] +/-  162ms
^- sth2-ts.nts.netnod.se       1   8   377    91  -3843us[-3843us] +/-  171ms
^* time.cloudflare.com         3   9   377   230    +13ms[  +13ms] +/-  129ms
^+ virginia.time.system76.c>   2   9   377   226  -8753us[-8753us] +/-  116ms
```

3. Manually synchronize the time using the **# chronyc makestep** command.

NOTE

For additional details on Chrony, check the following links:
https://linux.die.net/man/8/chronyd
https://linux.die.net/man/1/chronyc

Setting the Time Zone

There are two ways to configure the Moxa Arm-based computer's time zone. One is using the **TZ** variable. The other is using the **/etc/localtime** file.

Using the TZ Variable

The format of the TZ environment variable looks like this:

TZ=*<Value>HH[:MM[:SS]][daylight[HH[:MM[:SS]]][,start date[/starttime], enddate[/endtime]]]*

Here are some possible settings for the North American Eastern time zone:

1. **TZ=EST5EDT**
2. **TZ=EST0EDT**
3. **TZ=EST0**

In the first case, the reference time is GMT, and the stored time values are correct worldwide. A simple change of the TZ variable can print the local time correctly in any time zone.

In the second case, the reference time is Eastern Standard Time and the only conversion performed is for Daylight Saving Time. Therefore, there is no need to adjust the hardware clock for Daylight Saving Time twice per year.

In the third case, the reference time is always the time reported. You can use this option if the hardware clock on your machine automatically adjusts for Daylight Saving Time or you would like to manually adjust the hardware time twice a year.

```
moxa@Moxa-tbzkb1090923:~$ TZ=EST5EDT
moxa@Moxa-tbzkb1090923:~$ export TZ
```

You must include the TZ setting in the **/etc/rc.local** file. The time zone setting will be activated when you restart the computer.

The following table lists other possible values for the TZ environment variable:

Hours From Greenwich Mean Time (GMT)	Value	Description
0	GMT	Greenwich Mean Time
+1	ECT	European Central Time
+2	EET	European Eastern Time
+2	ART	
+3	EAT	Saudi Arabia
+3.5	MET	Iran
+4	NET	
+5	PLT	West Asia
+5.5	IST	India
+6	BST	Central Asia
+7	VST	Bangkok
+8	CTT	China
+9	JST	Japan
+9.5	ACT	Central Australia
+10	AET	Eastern Australia
+11	SST	Central Pacific
+12	NST	New Zealand
-11	MIT	Samoa
-10	HST	Hawaii
-9	AST	Alaska
-8	PST	Pacific Standard Time
-7	PNT	Arizona
-7	MST	Mountain Standard Time
-6	CST	Central Standard Time
-5	EST	Eastern Standard Time
-5	IET	Indiana East
-4	PRT	Atlantic Standard Time
-3.5	CNT	Newfoundland
-3	AGT	Eastern South America
-3	BET	Eastern South America
-1	CAT	Azores

Using the localtime File

The local time zone is stored in the **/etc/localtime** and is used by GNU Library for C (glibc) if no value has been set for the TZ environment variable. This file is either a copy of the **/usr/share/zoneinfo/** file or a symbolic link to it. The Arm-based computer does not provide **/usr/share/zoneinfo/** files. You should find a suitable time zone information file and write over the original local time file on the Arm-based computer.

4. Using and Managing Computer Interfaces

In this chapter, we include more information on the Arm-based computer's interfaces, such as the serial interface, storage, and the diagnostic LEDs. The instructions in this chapter cover all functions supported in Moxa's Arm-based computers. Before referring to the sections in this chapter, make sure that they are applicable to and are supported by the hardware specification of your Arm-based computer.

Moxa Computer Interface Manager (MCIM)

On many occasions, there isn't one standard method to access and configure specific interfaces on Moxa Arm-based computers because the hardware varies. Hence, programing across different Moxa Arm-based computer models can be difficult and time consuming. The goal of MCIM is to provide a unified software interface to access and configure non-standard computer interfaces. For example, MCIM can change the serial port interface mode (e.g., RS-232, RS-485-2W, RS-422). However, configuring the serial port baud rate is not possible in MCIM because Linux provides a standard method to set the baud rate.

MCIM is a command-line interface (CLI) Moxa utility designed to access and manage Moxa Arm-based computers' interfaces. Use the **`# mx-interface-mgmt`** command to display the menu page.

Configuring the Log Level

To set the log level of MCIM, edit the configuration file
`/etc/moxa/MoxaComputerInterfaceManager/MoxaComputerInterfaceManager.con`

Key	Value	Description
LOG_LEVEL	debug/info/warn/error	The log-level settings for the logs generated by MCIM for debugging and troubleshooting. The default level is "info"

Device Information

Use the **`# mx-interface-mgmt deviceinfo`** command to get information on your Moxa Arm-based computer.

Command and Usage	Description
deviceinfo	Show the following information: • Serial number (S/N) • Model name. • SECUREBOOT (Enabled / Disabled)

LED Indicators

Use **# mx-interface-mgmt led** command to get the list of controllable LEDs on your Arm-based computer. In the following example, the returned NAME RDY" refers to the green LED for the system ready, labeled "RDY" on the device. For **LEDs** with **multiple colors** such as U1 (red and green), 2 LED names will appear (U1_Red and U1_Green). For this type of LEDs, you must set the state of a color to "off" before setting another color to "on" or "blinking".

```
moxa@moxa-tbckb1045627:~$ mx-interface-mgmt led
NAME       LABEL                  STATE  ALIAS
RDY        RDY:green:status       on     N/A
RDY_Red    RDY:red:status         off    N/A
SD         SD:green:status        off    N/A
U1_Green   U1:green:programming   off    N/A
U1_Red     U1:red:programming     off    N/A
U2_Green   U2:green:programming   off    N/A
U2_Red     U2:red:programming     off    N/A
```

The MCIM commands for LED indicator controls are listed in the following table:

Command and Usage	Description
led	Shows the following information for all controllable LEDs • Name (as labeled on the device) • Model series of the device. • Color of the LED. • Description of the LED. • LED state (on/off/heartbeat)
led *<led_name>*	Show the above information of a **specified** LED
led *<led_name>* get_state	Get the current state (on/off/heartbeat) of a **specified** LED
led *<led_name>* set_state *<led_state>*	Set the state of a **specified** LED. Value of <state> can be **on**, **off**, or **heartbeat**

An example of changing the current state of U1 LED from **red** (steady) to red (heartbeat) is given below:

```
moxa@moxa-tbckb1045627:~$ mx-interface-mgmt led U1_Red
NAME=U1_Red
LABEL=U1:red:programming
STATE=off
ALIAS=N/A
moxa@moxa-tbckb1045627:~$ sudo mx-interface-mgmt led U1_Red set_state heartbeat
moxa@moxa-tbckb1045627:~$ sudo mx-interface-mgmt led U1_Red get_state
heartbeat
```

Storage and Partitions

Use **# mx-interface-mgmt disk and # mx-interface-mgmt partition** commands for managing the storage device and partitions.

Command and Usage	Description
disk	Show the following information of **all** embedded and external storage • Name (e.g., eMMC, SD) • Device node (e.g., /dev/mmcblk0) • System disk (Y/N), if 'Y', it is the disk with MIL installed. • Number of partitions. • Automount enabled/disabled (Y/N)
disk <*disk_name*>	Show the following information of a **specified** storage device • Name (e.g., eMMC, SD) • Device node (e.g., /dev/mmcblk0) • System disk (Y/N), if 'Y', it is the disk with MIL installed. • Partition name and device node. • Automount enabled/disabled (Y/N)
disk <*disk_name*> set_automount <*value*>	Set a **specified** external storage device (e.g., SD) to automount when attach to device; <value> is true/false
partition	Show the following information for partitions on **all** embedded and external storage devices: • Name (e.g., eMMC_p1, eMMC_p2) • Device node (e.g., /dev/mmcblk0p1) • Partition mounted (Y/N) • Partition mount point (e.g., /boot_device/p1) • Filesystem (e.g., ext4, FAT32)
partition <*partition_name*>	Show the above information of a **specified** partition
partition <*partition_name*> mount	Mount a **specified** partition
partition <*partition_name*> unmount	Unmount a **specified** partition

For example, to query available storage device and set SD storage drive to automount, use the following command:

```
NAME  DEVICE        SYSTEM_DISK  NUMBER_OF_PARTITIONS  AUTOMOUNT_SETTING
IO_STATE
SD    /dev/mmcblk0  N            1                     false              N/A
eMMC  /dev/mmcblk2  Y            4                     false              N/A
moxa@moxa-tbckb1045627:~$ # mx-interface-mgmt disk SD set automount true
```

To query available partitions and mount the partition 1 of the SD storage drive, use the following command:

```
moxa@moxa-tbckb1045627:~$ mx-interface-mgmt partition
NAME     DEVICE          IS_MOUNTED  FS_TYPE  MOUNTPOINT       MAPPER_DEVICE
SD_p1    /dev/mmcblk0p1  N           N/A      N/A              N/A
eMMC_p1  /dev/mmcblk2p1  Y           ext4     /boot_device/p1  N/A
eMMC_p2  /dev/mmcblk2p2  Y           ext4     /boot_device/p2  N/A
eMMC_p3  /dev/mmcblk2p3  Y           ext4     /boot_device/p3  N/A
eMMC_p4  /dev/mmcblk2p4  Y           ext4     /boot_device/p4  N/A

moxa@moxa-tbckb1045627:~$ sudo mx-interface-mgmt partition SD_p1 mount
moxa@moxa-tbckb1045627:~$ sudo mx-interface-mgmt partition SD_p1
NAME=SD_p1
DEVICE=/dev/mmcblk0p1
IS_MOUNTED=Y
FS_TYPE=vfat
MOUNTPOINT=/media/SD_p1
MAPPER_DEVICE=N/A
UUID=4C85-5936
```

WARNING

Setting external storage device to automount may expose your device to cybersecurity risks. It is strongly recommended that you do not automount storage device unless your device is placed is in a highly secure environment.

Serial Port

The serial ports support RS-232, RS-422, and RS-485 2-wire operation modes with flexible baudrate settings. The default operation mode is RS-232.

Use the **`# mx-interface-mgmt serialport`** command to query and configure the operation mode for the serial ports.

Command and Usage	Description
`serialport`	Shows the following information for **all** serial ports on the device: • Name (as labeled on device) • Device node (e.g., /dev/ttyM0)
`serialport <serialport_name>`	Shows the following information for a **specified** serial port: • Name (as labeled on device) • Device node (e.g., /dev/ttyM0) • Supported operation modes (e.g., RS-232, RS-485-2W, RS-422) • Supported baudrates. • Current operation mode configured.
`serialport <serialport_name> get_interface`	Gets the current operation mode for a **specified** serial port
`serialport <serialport_name> set_interface <serial_interface>`	Sets the operation mode for a **specified** serial port.

Changing the Serial Port Operation Mode

For example, to change the mode of P1 serial port from default RS-232 mode to the RS-485-2W mode, use the following command:

```
moxa@moxa-tbckb1045627:~$ mx-interface-mgmt serialport
NAME  DEVICE      INTERFACE  IO_STATE
P1    /dev/ttyM0  RS-232     N/A
P2    /dev/ttyM1  N/A        N/A
moxa@moxa-tbckb1045627:~$ mx-interface-mgmt serialport P1
NAME=P1
DEVICE=/dev/ttyM0
SUPPORTED_INTERFACES=RS-232,RS-485-2W,RS-422
SUPPORTED_BAUDRATES=300,600,1200,1800,2400,4800,9600,19200,38400,57600,115200
INTERFACE=RS-232
moxa@moxa-tbckb1045627:~$ sudo mx-interface-mgmt serialport P1 set_interface
RS-485-2W
moxa@moxa-tbckb1045627:~$ mx-interface-mgmt serialport P1 get_interface
RS-485-2W
```

For ioThinx 4533, it only supports following serial combinations for P1 and P2:

	#1	#2	#3
P1	**RS-232**	**RS-422**	**RS-485-2W**
P2	**N/A**	**N/A**	**RS-485-2W**

Changing Other Serial Interface Settings with STTY

The **stty** command is used to view and modify the serial terminal settings.

Displaying All Settings

Use the following example to display all serial terminal settings of COM1 serial port.

```
moxa@moxa-tbckb1045627:~$ mx-interface-mgmt serialport
NAME  DEVICE      INTERFACE  IO_STATE
P1    /dev/ttyM0  RS-485-2W  N/A
P2    /dev/ttyM1  N/A        N/A
moxa@moxa-tbckb1045627:~$ sudo stty -a -F /dev/ttyM0
speed 9600 baud; rows 0; columns 0; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; swtch = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R;
werase = ^W; lnext = ^V; discard = ^O; min = 1; time = 0;
-parenb -parodd -cmspar cs8 hupcl -cstopb cread clocal -crtscts
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon -ixoff
-iuclc -ixany -imaxbel -iutf8
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprt
echoctl echoke -flusho -extproc
```

Configuring Serial Settings

The following example changes the baudrate to 115200.

```
moxa@moxa-tbckb1045627:~$ sudo stty 115200 -F /dev/ttyM0
```

Check the settings to confirm that the baudrate has changed to 115200.

```
moxa@moxa-tbckb1045627:~$ sudo stty -a -F /dev/ttyM0
speed 115200 baud; rows 0; columns 0; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; swtch = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R;
werase = ^W; lnext = ^V; discard = ^O; min = 1; time = 0;
-parenb -parodd -cmspar cs8 hupcl -cstopb cread clocal -crtscts
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon -ixoff
-iuclc -ixany -imaxbel -iutf8
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprt
echoctl echoke -flusho -extproc
```

NOTE

Detailed information on the **stty** utility is available at the following link:

https://manpages.debian.org/bullseye/coreutils/stty.1.en.html

Ethernet Interface

Use **# mx-interface-mgmt ethernet** command to display information on the Ethernet ports.

Command and Usage	Description
ethernet	Show the following information of **all** ethernet ports on the device. • Name (as labeled on device) • Network interface name (eth0, eth1, etc.)
ethernet <ethernet_name>	Show the above information of a **specified** ethernet port

```
moxa@moxa-tbckb1045627:/$ mx-interface-mgmt ethernet
NAME   DEVICE_NAME   IO_STATE
L1     eth0          N/A
L2     eth1          N/A
moxa@moxa-tbckb1045627:/$ mx-interface-mgmt ethernet L1
NAME=L1
DEVICE_NAME=eth0
IO_STATE=N/A
```

Serial Console Interface

Use the **# mx-interface-mgmt console** command to display the serial console port information.

Command and Usage	Description
console	Show the following information for the console port. • Name (as labeled on the device) • Device node (e.g., /dev/ttymxc0)
Console <console_name>	Show the above information of a specified serial console interface

Following is an example of showing the console port device node.

```
moxa@moxa-tbckb1045627:/$ mx-interface-mgmt console
NAME      DEVICE         IO_STATE
Console   /dev/ttymxc0   N/A
```

Push-button

Getting the Button List and Status

Use **# mx-interface-mgmt button** command to display the available buttons and the button-configured actions.

Command and Usage	Description
button	Show the following information for **all** buttons on the device: • Name (as labeled on device) • Action (default/user-defined/disabled) ➢ Default: Button behavior is default ➢ User-defined: The button behavior has been customized by the user ➢ Disabled: The button has no function when pushed.
button <name>	Show the above information for a **specified** button

Following is an example of using MCIM to query an available button (Reset button) of the ioThinx 4533.

```
moxa@moxa-tbckb1045627:/$ mx-interface-mgmt button
NAME   ACTION
FN     default
```

Customize the Button Action

You can use the two scripts (default and custom) available in the following path to customize button actions: **/etc/moxa/MoxaComputerInterfaceManager/button-scripts/**. For example, in the ioThinx 4533, the default script is "**ioThinx4530-default.script**" and custom script is "**custom.script**".

By default, the FN button will load the default script when pressed. The default script will perform designed tasks based on the actions of the FN button. The following table gives a detailed description of the default script:

FN button Action	LED Indicator Status	Resulting Action
Press and hold FN button and release within 1s	SYS LED blinks	Device reboot
Press and hold FN button and release between 7s to 9s	• SYS LED blinks for 1s to 6s. • SYS LED is ON for 7s to 9s.	Reset to factory default
Press and hold FN button and release after 9s	• SYS LED blinks for 1s to 6s. • SYS LED is ON for 7s to 9s. • SYS LED is OFF after 9s.	Do nothing; cancel action

To customize the **FN** button action, a configuration file at **/etc/moxa/MoxaComputerInterfaceManager/peripheral-settings.conf** could be modified. The device needs to reboot for the settings to take effect.

The **Action** parameter in the configuration file can have the following three values:

- 0: Disable the button (no action when pressed)
- 1: Run the default script
- 2: Run the custom script

NOTE

You must reboot the system for the settings to take effect.

An example of the settings in the **peripheral-settings.conf** file is shown below:

```
[Disk/eMMC]
AutoMount=false

[Disk/SD]
AutoMount=false

[Button/FN]
Action=2

[SerialPort/P2]
Interface=1

[SerialPort/P1]
Interface=0
```

If **Action** is set to 2 (custom script), **/etc/moxa/MoxaComputerInterfaceManager/button-scripts/custom.script** should be edited to add the desired actions. To make it easier to configure the actions in the script file, copy the content of the default script to custom script file and then make the required changes.

```
root@moxa-tbzkb1090923:/etc/moxa/MoxaComputerInterfaceManager/button-scripts#
cp ioThinx4530-default.script custom.script
```

5. Configuring and Managing Networks

Configuring the Ethernet Interfaces

After the first login, you can configure the ioThinx 4533 controller's network settings to fit your application better. Note that it is more convenient to manipulate the network interface settings from the serial console than from an SSH login because an SSH connection can disconnect when there are network issues and the connection must be reestablished.

Modifying Network Settings via the Serial Console

In this section, we use the serial console to configure the ioThinx 4533 controller's network settings. Follow the instructions in the Connecting to the ioThinx 4533 controller section under Getting Started to access the Console Utility of the target computer via the serial Console port and then type cd /etc/network to change directories.

```
moxa@moxa-tbckb1045627:/$ cd /etc/network
moxa@moxa-tbckb1045627:/etc/network$
```

Type sudo vi interfaces to edit the network configuration file in the vi editor. You can configure the ioThinx 4533 controller's Ethernet ports to use either static or dynamic (DHCP) IP addresses.

Setting a Static IP address

To set a static IP address for the ioThinx 4533 controller, use the iface command to modify the default gateway, address, network, netmask, and broadcast parameters of the Ethernet interface.

```
moxa@moxa-tbckb1045627:/etc/network$ sudo vi interfaces

# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d
auto eth0 eth1 lo
iface lo inet loopback
iface eth0 inet static
        address 192.168.3.127
        network 192.168.3.0
        netmask 255.255.255.0
        broadcast 192.168.3.255
iface eth1 inet static
        address 192.168.4.127
        network 192.168.4.0
        netmask 255.255.255.0
        broadcast 192.168.4.255
```

Setting Dynamic IP Addresses:

To configure one or both LAN ports to request an IP address dynamically use the dhcp option in place of static in the iface command. Changing the LAN1(eth0) to DHCP, for example, as follows:

```
# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d
auto eth0 eth1 lo
iface lo inet loopback
iface eth0 inet dhcp
iface eth1 inet static
        address 192.168.4.127
        network 192.168.4.0
        netmask 255.255.255.0
        broadcast 192.168.4.255
```

6. System Installation and Update

In this chapter, we will introduce how to install and update Moxa Industrial Linux and the bootloader.

Installing Moxa Industrial Linux

Using a TFTP Server From Bootloader Menu

Refers to instruction in Accessing Bootloader Menu section.

Using an SD From Bootloader Menu

Refers to instruction in Accessing Bootloader Menu section.Automatic Installation From a SD

Besides manually installing the system image from bootloader menu, you can also trigger the image installation process within the operating system using **`mx-bootloader-mgmt image_auto_install`** command. Once this process is triggered, the Arm-based computer will automatically install the specified system image in the SD attached to the system. The new image will be available upon the next system boot-up.

NOTE

The format supported for SD is FAT32 and ext4, respectively.

Command	Description
-d, --disk	Display the name of the external storage (e.g., SD) where the image file is located. You can use the **`mx-interface-mgmt disk`** command to query the external storage name.
-f, --file	Display the name of the image file in the external storage
-i, --info	Display the names of the image file and external storage configured for auto-install upon next boot-up
-r, --remove	Remove the auto-installation configuration
-h, --help	Display the available commands with a brief description
-v, --version	Display the version of **`mx-image-auto-install-tool`**

Following is an example of the automatic installation of the system image from a SD device:

1. Use **`mx-interface-mgmt disk`** command to check the name of available storage device name.

```
moxa@moxa-tbckb1045627:~$ mx-interface-mgmt disk
NAME   DEVICE        SYSTEM_DISK  NUMBER_OF_PARTITIONS  AUTOMOUNT_SETTING
SD     /dev/mmcblk0  N            1                     false
eMMC   /dev/mmcblk2  Y            4                     false
```

2. Mount the SD if it is not already mounted. Refer to Storage and Partition section for detail.

```
moxa@moxa-tbckb1045627:~$ sudo mx-interface-mgmt partition
NAME      DEVICE          IS_MOUNTED  FS_TYPE  MOUNTPOINT
SD_p1     /dev/mmcblk0p1  N           N/A      N/A
eMMC_p1   /dev/mmcblk2p1  Y           ext4     /boot_device/p1
eMMC_p2   /dev/mmcblk2p2  Y           ext4     /boot_device/p2
eMMC_p3   /dev/mmcblk2p3  Y           ext4     /boot_device/p3
eMMC_p4   /dev/mmcblk2p4  Y           ext4     /boot_device/p4
```

3. Configure an auto-installation event in partition 1 of the SD device with the image file **IMG_ioThinx-4530_MIL3IOTHINX_2.0.0_Build_24011609_ImageBuild_240116_145008.img**:

```
moxa@moxa-tbckb1045627:~$ sudo mx-bootloader-mgmt image_auto_IMG_ioThinx-
4530_MIL3IOTHINX_2.0.0_Build_24011609_ImageBuild_240116_145008.img240116_145
008.img
Image auto install configuration:
Image File: IMG_ioThinx-
4530_MIL3IOTHINX_2.0.0_Build_24011609_ImageBuild_240116_145008.img
Disk Name: SD
```

NOTE

Ensure that the image file and sha256 hash files are available in partition 1 of SD before configuring the event.

4. Reboot the system to trigger the auto installation of the system image from the SD card.

```
moxa@moxa-tbzkb1090918:~# sudo reboot
```

7. Backup, Decommission, and Recovery

In this chapter, we will introduce how to use Moxa System Management (MSM) utility to perform snapshot, backup, decommission of your system.

Function	Description
Snapshot	• The snapshot has a smaller footprint as it saves just the differences (partition 3 in Figure 7.1) compared to the out-of-factory rootfs (partition 2 in Figure 7.1) as well as the Linux Kernel (partition 1 in Figure 7.1). • The snapshot is saved in the Moxa Arm-based computer and cannot be exported. Hence, a snapshot can only be used to restore the computer that the snapshot was taken from.
Backup	• The backup has a larger footprint as it saves the entire system, including the out-of-factory rootfs. • The backup can be exported to an external storage. • The backup can be used to restore the Moxa Arm-based computer that the backup is taken from or another computer of the same model.

Below diagram illustrate an overview of MIL3 system layout:

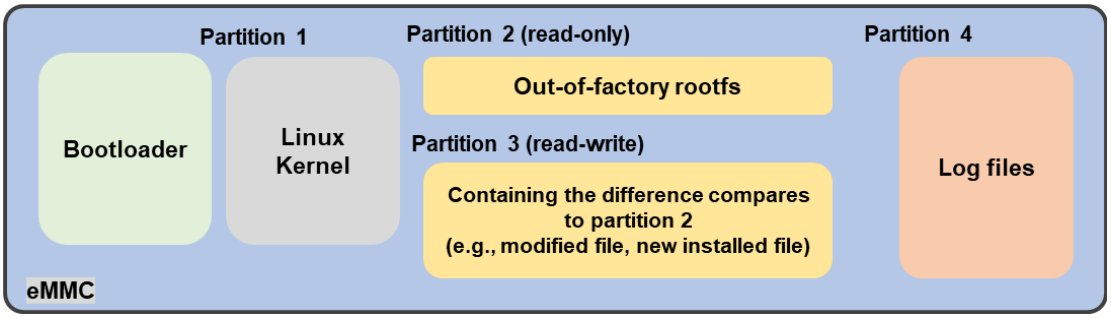


Figure 7.1 - Layout Overview of ioThinx 4533 with MIL3

Creating a System Snapshot

A snapshot preserves the state and data of the Moxa Arm-based computer as a restoration point at a specific point in time so that you can restore it to that point if something goes wrong. Snapshots only save the Linux kernel and new and modified files to the out-of-factory rootfs (partition 2). Therefore, the size of a snapshot is much smaller than a backup.

Use the **`# mx-system-mgmt snapshot <sub-command> <flag>`** to create restore a system. You must use **`sudo`** or run the command with root permission.

Sub-commands	Description
create	Creates a snapshot of the system. • A snapshot includes **kernel (partition 1)** and **rootfs (partition 3)** • Only one snapshot is saved. A new snapshot will overwrite the previous snapshot. • Snapshot is stored in **rootfs (partition 3)**
restore	Restores the system with the snapshot.
delete	Deletes the existing snapshot.
Info	Displays the create time and size of the existing snapshot.

Flag	Description
-y or --yes	Automatically consent to the prompts during create, restore, and delete processes

Creating a System Backup

Compared to snapshot, a backup saves Linux kernel and the rootfs on your Moxa Arm-based Computer. Therefore, a backup can be exported and used to restore a Moxa Arm-based computer of the same model with MIL 3.0. For example, if you create a backup on ioThinx 4533 with MIL3, you can use the backup to restore another ioThinx 4533 with MIL3.

Use **`# mx-system-mgmt backup <sub-command> <flag>`** command to create, delete, and restore a backup. You must use **`sudo`** or run the command with the root permission.

Sub-commands	Description
create	Creates a backup of the system • The backup includes **kernel (partition 1), rootfs (partition 2)**, and **rootfs (partition 3)** • By default, the backup is created in the **/boot_device/p3/backup/** directory with the name **backup.tar**, together with an info file that contains the backup information and cryptographic hash of the backup. • The backup includes a system snapshot. If you would like to reduce the size of backup, you can delete the snapshot in the system before performing the backup if the snapshot is not needed.
delete	Deletes the backup from default directory
restore	Restores the system using the backup from default directory. • Existing snapshot on system will be deleted after restoring the system from a backup. • The cryptographic hash in the **info** file will be used to validate the integrity of the backup file before the restore process begins. • A system reboot is required after restoration.
info	Displays the creation time and size of the backup in the default directory
-D or --directory	Specifies the directory for **`create`**, **`delete`**, **`restore`** and **`info`** commands

Flag	Description
-y or --yes	Automatically consent to the prompt during create, delete, and restore

The following example shows how to back up a system to aSD card with the mounting point.

/media/SD_p1:

```
moxa@moxa-tbckb1045627:/$ sudo mx-system-mgmt backup create -D /media/SD_p1
Set /media/SD_p1 as backup directory.
Check the backup information...
Type: backup
Create Time: 2024.03.17-13:45:55
Size: 333MB
A backup already exist. This will overwrite the existing backup.
Would you like to continue? (y/N)
y
Start evaluating space, please wait...
Estimation of Required Space: 332MB
Available Space: 6407MB
Would you like to continue? (y/N)
y
Synchronize boot files...
              0   0%    0.00kB/s    0:00:00 (xfr#0, to-chk=0/2)
Start creating backup file...
There is no /boot_device/p2/rootfs.sqfs.sha256sum.bin.signed file, the system
environment is insecure.
 332MiB 0:00:31 [10.7MiB/s]
[                   <=>                                 ]
Type: backup
Create Time: 2024.03.17-13:53:37
Size: 333MB
The backup has been created successfully under: /media/SD_p1
```

The following example shows how to restore a backup from the SD card with the mounting point.

/media/SD_p1:

```
moxa@moxa-tbckb1045627:/$ sudo mx-system-mgmt backup restore -D /media/SD_p1
Set /media/SD_p1 as backup directory.
Check the backup information...
Type: backup
Create Time: 2024.03.17-13:53:37
Size: 333MB
Start verifying backup file, please wait...
Verified OK!
Start evaluating space, please wait...
Estimation of Required Space: 333MB
Available Space: 5485MB
Would you like to continue? (y/N)
Y
Check the snapshot information...
There is no snapshot information
To restore the backup file will overwrite current system and factory default
system.
Do you want to continue? (y/N)
Y
Start using the backup file to restore the system...
 332MiB 0:00:08 [40.3MiB/s] [============================================>]
100%
There is no /boot_device/p2/rootfs.sqfs.sha256sum.bin.signed file, the system
environment is insecure.
Synchronize boot files...
              0   0%    0.00kB/s    0:00:00 (xfr#0, to-chk=0/2)
System has been restored successfully. Reboot is required to take effect.
moxa@moxa-tbckb1045627:~$ sudo reboot
```

Setting the System to the Default

Press and hold the **Reset** button for 7 to 9 seconds to reset the computer to the factory default settings. When the reset button is held down, the RDY LED will blink once every second. The LED will become steady when you hold the button continuously for 7 to 9 seconds. Release the button immediately when the LED becomes steady to load the factory default settings. For additional details on the LEDs, refer to the quick installation guide or the user's manual for ioThinx 4533.

ATTENTION

Reset-to-default will erase all data stored in the boot-up storage

Back up your files before resetting the system to factory defaults. All the data stored in the Arm-based computer's boot-up storage will be destroyed after resetting to factory defaults.

You can also use the **`mx-system-mgmt default restore`** command to restore the computer to factory default settings. You must use sudo or run the command with the root permission.

```
moxa@moxa-tbzkb1090923:/# sudo mx-system-mgmt default restore
```

If you would like to configure the **Reset** button for a different action (e.g., restore to a snapshot), refer to Customize the Button Action section.

Decommissioning the System

Compared with the set-to-default function, decommissioning will further erase all data stored in the log partition to help erase security-sensitive information.

ATTENTION

Decommission will erase all the data, including event and audit logs

Please back up your files before resetting the system to factory defaults. All user data, including logs in your Arm-based computer, will be destroyed after decommissioning. Bootloader configuration, including administrator password, will also be set to factory default.

You can also use the **`mx-system-mgmt default decommission`** command to restore the computer to factory default. You must use sudo or run the command with the root permission.

```
moxa@moxa-tbzkb1090923:/# sudo mx-system-mgmt default decommission
```

The decommissioning process will do the following:

1. Overwrite the system partition 4 times with shred so that all user files will be deleted and cannot be recovered.
2. Overwrite the log partition 4 times with shred so that all log files will be deleted and cannot be recovered.
3. Trigger the bootloader decommissioning function, so all configurations and log messages in the bootloader are also deleted and cannot be recovered.

8. Security Capability

In this chapter, we will introduce Moxa Arm-based computers' key security functions and a security hardening guide to deploy and operate Moxa computer in a secure manner.

Communication Integrity and Authentication

Below is a list of network communication services and protocols available in the Moxa Arm-based computer and their data integrity and authentication protection mechanisms.

Service	Protocol	Data Integrity	Data Authentication
SSH server and client	SSH	HMAC algorithm is used to guarantee data integrity	Uses key signature algorithms such as ED25519, ECDSA, or RSA to verify authenticity.
SFTP server	SSH		
SCP server	SSH		
APT client	HTTPS	SecureAPT uses checksum to guarantee data integrity	SecureAPT uses GPG public key system to validate data authenticity
NTP client (NTS support)	TLS/SSL, NTP	NTS guarantees data integrity via NTS Authenticator and Encrypted EF	NTS provides TLS layer to guarantee authenticity

ATTENTION

For post-installed communication services and protocols, you must ensure data integrity and authentication are implemented. If integrity and authentication are not available, you must use additional compensating countermeasures in the system to compensate the risk. For example, physical cable protection for serial Modbus RTU.

User Account Permissions and Privileges

Switching to the Root Privilege

In Moxa Arm-based computers, the root account is disabled in favor of better security. The default user account **moxa** belongs to the sudo group. Sudo is a program designed to let system administrators allow permitted users to execute some commands as the root user or another user. The basic philosophy is to give as few privileges as possible, but still allow people to get their work done. Using sudo is better (safer) than opening a session as a root for several reasons, including:

- Nobody needs to know the root password (sudo prompts for the current user's password). Extra privileges can be granted to individual users temporarily, and then taken away with no password change.
- It is easy to run only the commands that require special privileges via sudo; the rest of the time, you work as an unprivileged user, which reduces the damage caused by mistakes.
- Some system-level commands are not available to the user moxa directly, as shown in the sample output below:

```
moxa@Moxa-tbzkb1090923:~$ sudo ifconfig
eth0      Link encap:Ethernet  HWaddr 00:90:e8:00:00:07
          inet addr:192.168.3.127  Bcast:192.168.3.255  Mask:255.255.255.0
          UP BROADCAST ALLMULTI MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
```

```
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth1      Link encap:Ethernet  HWaddr 00:90:e8:00:00:08
          inet addr:192.168.4.127  Bcast:192.168.4.255  Mask:255.255.255.0
          UP BROADCAST ALLMULTI MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:32 errors:0 dropped:0 overruns:0 frame:0
          TX packets:32 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:2592 (2.5 KiB)  TX bytes:2592 (2.5 KiB)
```

You can switch to the root account using the **sudo -i (or sudo su)** command. For security reasons, do not operate all commands from the root account.

NOTE

Click the following link for more information on the **sudo** command.
https://wiki.debian.org/sudo

ATTENTION

You might get the permission denied message when using pipe or redirect behavior with a non-root account.

You must use '**sudo su -c**' to run the command instead of using >, <, >>, <<, etc.

Note: The single quotes enclosing the full command are required.

Controlling Permissions and Privileges

Moxa Industrial Linux uses Discretionary Access Control (DAC) based on Access Control Lists (ACLs) to manage permissions and privileges, which an object has an owner that controls the permissions to access the object. Subjects can transfer their access to other subjects. In other words, the owner of the resource has full access and can determine the access type (rwx: read, write, execute) of other users.

You can use **chmod** command to configure who (user, group, other) can do what (read, write, execute) to a file or directory. The access permission is extended by Access Control Lists (ACLs) authorization. ACL provides a more flexible mechanism that allows multiple users and groups to own an object. You can check and configure access control lists of a specific file or directory using **getfacl** and **setfacl** commands.

NOTE

Click the following link for more information on usages of chmod and Access Control Lists (ACLs)
https://wiki.debian.org/Permissions

Moxa Arm-based computers only provide one account in sudo group by default because it is intended for the system integrator to customize and build their applications on top.

The system integrator shall be responsible for setting the appropriate permissions to roles and user accounts to enforce the concept of least privilege.

Linux Login Policy

Invalid Login Attempts

Moxa Industrial Linux provides the capability to configure allowed invalid login attempts to mitigate against Denial-of-Service (DoS) and Brute-force attack.

Security Model	Default Rule
Standard model	Not set

Following is the configuration file and variable to configure the setting:

Configuration Option	Configuration file	Variable to Set
Consecutive invalid login	/etc/security/faillock.conf	deny
Within how many seconds	/etc/security/faillock.conf	fail_interval
Deny access for how long (in seconds)	/etc/security/faillock.conf	unlock_time

More configurable options can be found in following reference:

- login.defs(5) — login — Debian bullseye — Debian Manpages
- faillock.conf(5) — libpam-modules — Debian bullseye — Debian Manpages

Session Termination After Inactivity

This setting automatically terminates the login sessions after a standard period of inactivity. Below is the default configuration set in Moxa Arm-based computer.

Security Model	Default Value
Standard model	Not set

Follow below instructions to configure the inactivity time:

Login Method	Configuration
Serial Console and SSH (Secure Shell)	• Set the value (in seconds) of variable **`TMOUT`** in **`/etc/profile.d/99-moxa-profile.conf`**. • Apply the same value to variable **`ClientAliveInterval`** in **`/etc/ssh/sshd_config.d/00-moxa-sshd.conf`**. • To apply the rule to sudo user, make sure variable **`env_keep+="TMOUT"`** exist in **`/etc/sudoers.d/00-moxa-sudoers-conf`**.

Login Banner Message

You can set a message banner message to display welcome or informational messages or warming message to un-authorized users. Follow the instructions below to add a banner Moxa Industrial Linux 3.0 UM for Arm-based Computers Moxa Industrial Linux 3.0 UM for Arm-based Computers.

Login Method	Banner Content	Additional Configuration Required
Serial Console	/etc/issue	n/a
SSH (Secure Shell)	/etc/issue.net	Add variable **`Banner /etc/issue.net`** is added in **`/etc/ssh/sshd_config.d/00-moxa-sshd.conf`**

Bootloader Login Policy

For bootloader login policy management, refers to the bootloader configuration section.

Trusted Platform Module (TPM 2.0)

The Moxa Arm-based computer includes a TPM 2.0 hardware module. TPM provides a hardware-based approach to manage user authentication, network access, data protection and more that takes security to higher level than software-based security. It is strongly recommended to manage keys with TPM and also store digital credentials such as passwords.

The TPM can be managed via the tpm2_tools pre-installed in Moxa Industrial Linux (https://github.com/tpm2-software/tpm2-tools).

TPM software stack & tool is maintained by tpm2-software community
https://tpm2-software.github.io/

A good reference of TPM 2.0 introduction
https://link.springer.com/chapter/10.1007/978-1-4302-6584-9_3

Intrusion Prevention

Fail2ban is pre-installed in Moxa Industrial Linux as an intrusion prevention software framework designed to prevent against brute-force attacks.

NOTE

Click the following link for detail instructions of Fail2ban usage
https://www.fail2ban.org/wiki/index.php/Main_Page

Network Security Monitoring

Zeek is pre-installed in Moxa Industrial Linux for network security monitoring. Zeek is a passive network traffic analyzer. Many operators use Zeek as a network security monitor (NSM) to support investigations of suspicious or malicious activity. Zeek also supports a wide range of traffic analysis tasks beyond the security domain, including performance measurement and troubleshooting. Zeek provides an extensive set of logs describing network activity. These logs include not only a comprehensive record of every connection seen on the wire but also application-layer transcripts.

You can enable Zeek to monitor the network traffic of these interfaces. Following the simple instruction below:

1. Export the Zeek environment.

```
export PATH=$PATH:/opt/zeek/bin
export ZEEK_PREFIX=/opt/zeek
```

2. [Required] Configure the interface to monitor by running **# vim $ZEEK_PREFIX/etc/node.cfg**.
3. [Required] Modify the interface list according to the interface you like to monitor. For example, add LAN1, LAN2 in the list.

```
# This example has a standalone node ready to go except for possibly
changing
# the sniffing interface.

# This is a complete standalone configuration.  Most likely you will
# only need to change the interface.
[zeek]
type=standalone
host=localhost
interface=eth0,eth1
```

4. [Optional] change the **MailTo** email address to a desired recipient and the **LogRotationInterval** to a desired log archival frequency

   vim $ZEEK_PREFIX/etc/zeekctl.cfg

```
# Recipient address for all emails sent out by Zeek and ZeekControl.
MailTo = root@localhost

# Rotation interval in seconds for log files on manager (or standalone)
node.
# A value of 0 disables log rotation.
LogRotationInterval = 3600
```

5. [Required] Run **$ZEEK_PREFIX/bin/zeekctl** to start **Zeek**

```
root@moxa-tbbbb1182827:/home/moxa# $ZEEK_PREFIX/bin/zeekctl

Hint: Run the zeekctl "deploy" command to get started.
Welcome to ZeekControl 2.4.0

Type "help" for help.
[ZeekControl] >
```

6. [Required] For the first-time use of the shell, use **install** command to perform initial installation of the ZeekControl configuration.

```
[ZeekControl] > install

creating policy directories ...
installing site policies ...
generating standalone-layout.zeek ...
generating local-networks.zeek ...
generating zeekctl-config.zeek ...
generating zeekctl-config.sh ...
[ZeekControl] >
```

7. [Required] Start Zeek instance by **Start** command (Use CTRL+D to exit if initializing successfully).

```
[ZeekControl] > start

starting zeek ...
(zeek still initializing)
```

8. View the Zeek logs under **$ZEEK_PREFIX/logs**.

```
root@moxa-tbbbb1182816:/# ls -alh /opt/zeek/logs/current/
total 96K
drwxr-sr-x 2 root zeek 4.0K Jun 19 04:18 .
drwxrws--- 1 root zeek 4.0K Jun 19 04:17 ..
-rw-r--r-- 1 root zeek  250 Jun 19 04:18 capture_loss.log
-rw-r--r-- 1 root zeek  128 Jun 19 04:17 .cmdline
-rw-r--r-- 1 root zeek  583 Jun 19 04:18 conn.log
-rw-r--r-- 1 root zeek  352 Jun 19 04:17 .env_vars
-rw-r--r-- 1 root zeek  30K Jun 19 04:17 loaded_scripts.log
-rw-r--r-- 1 root zeek  753 Jun 19 04:18 notice.log
-rw-r--r-- 1 root zeek  227 Jun 19 04:17 packet_filter.log
-rw-r--r-- 1 root zeek    5 Jun 19 04:17 .pid
-rw-r--r-- 1 root zeek   61 Jun 19 04:17 .startup
-rw-r--r-- 1 root zeek  686 Jun 19 04:17 stats.log
-rwxr-xr-x 1 root zeek   19 Jun 19 04:17 .status
-rw-r--r-- 1 root zeek   19 Jun 19 04:17 stderr.log
-rw-r--r-- 1 root zeek  204 Jun 19 04:17 stdout.log
-rw-r--r-- 1 root zeek  367 Jun 19 04:18 weird.log
```

NOTE

Click the following link for Zeek's detail instruction and the explanation on log types
https://docs.zeek.org/en/master/quickstart.html

If you prefer not to use ZeekControl (e.g., you don't need its automation and management features), you can refer to https://docs.zeek.org/en/master/quickstart.html#zeek-as-a-command-line-utility on how to directly control Zeek for your analysis activities from the command line for both live traffic and offline working from traces.

Firewall

nftable is the built-in firewall in Moxa Industrial Linux.

NOTE

Click the following link for detail instructions of nftable usages
https://wiki.nftables.org/wiki-nftables/index.php/Main_Page
https://wiki.nftables.org/wiki-nftables/index.php/Quick_reference-nftables_in_10_minutes

Pre-configured Rule

For ioThinx 4533, nftable is not enabled by default.

```
flush ruleset

table inet filter {
        chain input {
                type filter hook input priority 0;
        }
        chain forward {
                type filter hook forward priority 0;
        }
        chain output {
                type filter hook output priority 0;
        }
}
```

Common nftable Usage

1. List the currently loaded nftable rules **# nft list ruleset**
2. Debug and tracing if traffic are drop or accept as expected **# nft monitor trace**
   a. Add trace_chain before the existing input chain

   ```
   nft add chain inet filter trace_chain { type filter hook prerouting
   priority -1\; }
   ```

   b. Add nftrace flag

   ```
   nft add rule inet filter trace_chain meta nftrace set 1
   ```

   c. Monitor trace (you can use another device with ncat tool to test it)dd nftrace flag

   ```
   moxa@moxa-tbbbb1182816:/# sudo nft monitor trace

   trace id d51bda11 inet filter trace_chain packet: iif "eth0" ether saddr
   d8:5e:d3:a5:7b:29 ether daddr 00:90:e8:a6:37:cb ip saddr 192.168.1.102 ip
   daddr 192.168.1.107 ip dscp cs0 ip ecn not-ect ip ttl 128 ip id 36481 ip
   protocol tcp ip length 52 tcp sport 1142 tcp dport 53 tcp flags == syn
   tcp window 64240 trace id d51bda11 inet filter trace_chain rule meta
   nftrace set 1 (verdict continue) trace id d51bda11 inet filter
   ```

```
trace_chain verdict continue trace id d51bda11 inet filter trace_chain
policy accept trace id d51bda11 inet filter input packet: iif "eth0"
ether saddr d8:5e:d3:a5:7b:29 ether daddr 00:90:e8:a6:37:cb ip saddr
192.168.1.102 ip daddr 192.168.1.107 ip dscp cs0 ip ecn not-ect ip ttl
128 ip id 36481 ip protocol tcp ip length 52 tcp sport 1142 tcp dport 53
tcp flags == syn tcp window 64240 trace id d51bda11 inet filter input
verdict continue trace id d51bda11 inet filter input policy drop
```

d. Once debugging is completed, make sure to remove the debug flag by either method below:
   - ❒ Restart nftable **`# systemctl restart nftables`** or
   - ❒ Reload the configuration again **`# nft -f /etc/nftables.conf`**

Rate Limiting

Rate limiting is a common strategy to prevent network attacks such as DOS, DDOS, and brute force by limiting the network traffic within a specified time. As the suitable rate limit configuration depends heavily on the asset owner's applications, rate limiting is not configured by default in Moxa Industrial Linux.

nftable Rate Limit Usage	Example of Rate Limit Configuration
rate [over] <value> <unit> [burst <value> <unit>]	limit rate 400/minute limit rate 400/hour limit rate over 40/day limit rate over 400/week limit rate over 1023/second burst 10 packets limit rate 1025 kbytes/second limit rate 1023000 mbytes/second limit rate 1025 bytes/second burst 512 bytes limit rate 1025 kbytes/second burst 1023 kbytes limit rate 1025 mbytes/second burst 1025 kbytes limit rate 1025000 mbytes/second burst 1023 mbytes

You can directly add rate limit to existing rule in **`/etc/nftables.conf`**:

Below is an example of limiting TCP and UDP network traffic to 4 packets per second

```
#!/usr/sbin/nft -f

flush ruleset

define tcp_port_allow = { ssh, https };
define udp_port_allow = { 53, ntp };

table inet filter {
        # input: drop all traffic
        chain input {
                type filter hook input priority 0; policy drop;

                ct state invalid drop
                ct state established,related accept

                # allow icmp
                ip protocol icmp icmp type {
                        echo-request,
                        echo-reply,
                        time-exceeded,
                        parameter-problem,
                        destination-unreachable
                } accept

                # allow icmp6
                ip6 nexthdr icmpv6 icmpv6 type {
                        echo-request,
                        echo-reply,
                        time-exceeded,
```

```
                        parameter-problem,
                        destination-unreachable
                } accept

                # accept lo
iifname "lo" accept

                tcp dport $tcp_port_allow limit rate 4/second accept
                udp dport $udp_port_allow limit rate 4/second accept        }
```

Mitigating a NTP Amplification Attack

The default configured NTP servers in Moxa Industrial Linux (MIL) are with NTS support. If you use public NTP servers without NTS support, it is vulnerable to the **NTP amplification attack**, in which the attacker could exploit the public NTP servers to overwhelm Moxa Arm-based computer with UDP traffic. Under such an incident, you can follow the steps to stop the attack:

1. Stop NTP service temporarily with the # systemctl stop systemd-timesyncd command.
2. Block the tainted NTP server by nftables command:
   a. Create new firewall table
   ```
   nft add table inet firewall-filter
   ```
   b. Create new chain input in firewall table
   ```
   nft add chain inet firewall-filter input
   ```
   c. Create new chain input in firewall table
   ```
   nft add rule inet firewall-filter input tcp dport { ntp } ip saddr <your
   ip> reject
   ```
   d. Block NTP server IP
   ```
   nft add rule inet firewall-filter input tcp dport { ntp } ip saddr <your
   ip> reject
   ```
   e. Check the rule set
   ```
   nft list ruleset

   ...
   table inet firewall-filter {
           chain input {
                   tcp dport { 123 } ip saddr 10.213.123.55 reject
           }
   }
   ```
3. You can choose to specify another NTP server (modify **/etc/systemd/timesyncd.conf**) or wait for this server to finish troubleshooting.
4. Remember to flush the rule after recovery.
   ```
   nft delete chain inet firewall-filter input # delete chain
   # or
   nft delete table inet firewall-filter # delete table
   ```

Service and Ports

Only activate protocols you require using the system. Below is the list for the protocol and port numbers used for all external interfaces. Please refer to Firewall section to modify the list of allowed port if additional port is required.

Protocol	Protocol Type	Port Number
SSH	TCP	22
HTTPS	TCP	443
NTS	UDP	123/4460
DNS	UDP	53

Disable Unnecessary Protocols, Services, and Ports

You can use **#ss** to list all the current running processes using with the associated service, protocol, and network port.

```
moxa@moxa-tbbbb1182827:~$ sudo ss -tulpn
Netid       State         Recv-Q       Send-Q              Local Address:Port
Peer Address:Port       Process
tcp         LISTEN        0            128                       0.0.0.0:22
0.0.0.0:*           users:(("sshd",pid=974,fd=3))
tcp         LISTEN        0            128                          [::]:22
[::]:*            users:(("sshd",pid=974,fd=4))
```

You can disable a daemon or service by killing process ID (PID) directly. For example:

```
moxa@moxa-tbbbb1182827:~$ sudo kill 974
```

Or you can just stop and disable the service using **#systemctl**. For example:

```
moxa@moxa-tbbbb1182827:~$ sudo systemctl stop sshd
moxa@moxa-tbbbb1182827:~$ sudo systemctl disable sshd
```

Restrict Unnecessary Protocols, Services, and Ports

1. Protocols:

   Use **nfables** meta to match the kind of TCP traffic Matching packet metainformation. Refers to nftables wiki.

2. Services:

   Use **# systemctl list-unit-files** to find unused services and disable them by **systemctl disable <service>**.

3. Ports:

   Use nftables to add accepted ports in the whitelist. Refers to the Firewall section for detail instructions.

Services Enabled by Default

Below is the list for the services enabled by default in the ioThinx 4533.

Service Name	Description
chrony.service	chrony, an NTP client/server
dbus.service	D-Bus System Message Bus
fail2ban.service	Fail2Ban Service
getty@tty1.service	Getty on tty1
ifupdown-pre.service	Helper to synchronize boot up for ifupdown
kmod-static-nodes.service	Create list of static device nodes for the current kernel
ModemManager.service	Modem Manager
moxa-guardian.service	Moxa Guardian InitService
moxa-iothinx-io.service	ioThinx I/O Master
moxa-sys-rdy.service	Moxa system ready service
moxa-system-manager-init.service	Moxa System Manager Moxa system manager initial
moxa-system-manager.service	Moxa System Manager Moxa system manager
MoxaComputerInterfaceManager.service	Moxa Computer Interface Manager Moxa Computer Interface Manager
networking.service	Raise network interfaces
polkit.service	Authorization Manager
rsyslog.service	System Logging Service
serial-getty@ttymxc0.service	Serial Getty on ttymxc0
ssh.service	OpenBSD Secure Shell server
systemd-journal-flush.service	Flush Journal to Persistent Storage
systemd-journald.service	Journal Service
systemd-logind.service	User Login Management
systemd-modules-load.service	Load Kernel Modules
systemd-random-seed.service	Load/Save Random Seed
systemd-remount-fs.service	Remount Root and Kernel File Systems
systemd-sysctl.service	Apply Kernel Variables

Service Name	Description
systemd-sysusers.service	Create System Users
systemd-tmpfiles-setup-dev.service	Create Static Device Nodes in /dev
systemd-tmpfiles-setup.service	Create Volatile Files and Directories
systemd-udev-trigger.service	Coldplug All udev Devices
systemd-udevd.service	Rule-based Managerfor Device Events and Files
systemd-update-utmp.service	Update UTMP about System Boot/Shutdown
systemd-user-sessions.service	Permit User Sessions
user-runtime-dir@1000.service	User Runtime Directory /run/user/1000
user@1000.service	User Manager for UID 1000
vnstat.service	vnStat network traffic monitor
watchdog.service	watchdog daemon

Managing Resources

Setting The Process Priority

A process can be manually adjusted to increase or decrease its priority. Use the **top** or **ps** commands to find out the process priority.

```
moxa@moxa-tbckb1045627:~$ sudo top
top - 14:38:45 up 17 min,  1 user,  load average: 0.19, 0.14, 0.13

%Cpu(s):  0.7 us,  1.7 sy,  0.0 ni, 97.6 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   2009.2 total,   1854.1 free,     47.3 used,    107.8 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.   1900.0 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 1308 root      20   0    9944   2596   2104 R   2.7   0.1   0:00.62 top
  105 root      20   0       0      0      0 I   1.3   0.0   0:03.34 kworker/u+
   12 root      -2   0       0      0      0 I   0.7   0.0   0:00.26 rcu_preem+
   22 root      20   0       0      0      0 S   0.7   0.0   0:01.83 ksoftirqd+
   93 root     -51   0       0      0      0 S   0.7   0.0   0:05.28 irq/54-30+
  413 root       0 -20       0      0      0 I   0.7   0.0   0:00.17 kworker/0+
  530 message+  20   0    6288   3168   2576 S   0.7   0.2   0:01.93 dbus-daem+
  600 root      20   0   47896   6988   5280 S   0.7   0.3   0:03.38 MoxaCompu+

...
```

You can also use the **ps** command with the **-l**, long list option to find out the priority of the process.

```
moxa@moxa-tbckb1045627:~$ ps -l
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1000  1285   865  0  80   0 -  1952 do_wai ttymxc0  00:00:00 bash
0 R  1000  1312  1285  0  80   0 -  2396 -      ttymxc0  00:00:00 ps

...
```

The PRI (Priority) or NI (Nice) is the priority of the process. The PRI is adjusted by the kernel automatically. The NI can have a value in the range -20 to 19. A smaller value means that the program could use more CPU resources.

The nice utility can be given a specific nice value while running a program. This example shows how to launch the **tar** utility with the nice value 5.

```
moxa@moxa-tbbbb1182827:/# sudo nice -n 20 tar -czvf TheCompressFile.tar /src1
/src2 ...
OR
moxa@moxa-tbbbb1182827:/# sudo nice -adjustment 20 tar -czvf
TheCompressFile.tar /src1 /src2 ...
```

You can use the **renice** utility to dynamically adjust the nice value of a program. This example uses renice to adjust the auditd, PID 639, with highest priority as -20.

```
moxa@moxa-tbbbb1182827:/# sudo renice -n 20 -p 639
moxa@moxa-tbbbb1182827:/# sudo ps -efl|grep auditd
1 S root       639     1  0  75  -20 -  1519 poll_s 22:02 ?        00:00:00
/sbin/auditd -n
...
```

NOTE

Click the following link for more information on usages of nice and renice
https://manpages.debian.org/bullseye/coreutils/nice.1.en.html
https://manpages.debian.org/bullseye/bsdutils/renice.1.en.html

Setting the Process I/O Scheduling Class and Priority

The **ionice** command can adjust the priority of the program using I/O. The class and priority are adjustable for a process.

-c class	0: none 1: realtime 2: best-effort 3: idle
-n classdata	The realtime and best-effort can set from 0 to 7. A smaller value means the program has a higher priority.
-p PID	Process ID

```
moxa@moxa-tbbbb1182827:/# sudo ps -l
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S     0   895   886  0  80   0 -  1794 wait   pts/0    00:00:00 bash
4 S     0  1099   895  0  80   0 -  1659 poll_s pts/0    00:00:00 sudo
4 R     0  1100  1099  0  80   0 -  1850 -      pts/0    00:00:00 ps
moxa@moxa-tbbbb1182827:/# sudo ionice -c 2 -n 0 -p 895
moxa@moxa-tbbbb1182827:/# sudo ionice  -p 895
best-effort: prio 0
```

NOTE

Click the following link for more information on usages of ionice
https://manpages.debian.org/bullseye/util-linux/ionice.1.en.html

Limiting the CPU Usage of a Process Using cpulimit

cpulimit is a simple program that attempts to limit the CPU usage of a process (expressed in percentage, not in CPU time). This is useful to control batch jobs, when you don't want them to eat too much CPU.

This example, use the cpulimit to limit the usage of sshd process CPU limit percentage to 25% in background. The -p is the process ID. The -e switch takes the executable program file name. The -l is the CPU limit percentage. The option, -b, to run cpulimit in the background, freeing up the terminal.

```
moxa@moxa-tbbbb1182827:/# sudo cpulimit -p 895 -l 25 -b
```

NOTE

Click the following link for more information on usages of cpulimit
https://manpages.debian.org/bullseye/cpulimit/cpulimit.1.en.html

Limiting the Rate

Refer to the Chapter 8 Security Firewall Rate Limiting to customize the network limitation of the firewall configuration.

Audit Log

In this section, we will introduce the audit event log design in Moxa Industrial Linux and bootloader, including the security event monitored and recommended response and approach for audit processing failures.

Linux Audit log

Auditd is being used in Moxa Industrial Linux for system administrators to monitor detailed information about system operation. It provides a way to track and record security-relevant information on the system. For ioThinx 4533, the audit service is disabled by default.

1. Log partition size: **256MB**
2. Log partition applies Linux Unified Key Setup (LUKS) encryption and restricts non-root users from access
3. Logs are stored under **`/var/log/audit/`** and the log format follows **`auditd`** standard
   - Below is a reference of where to find the commonly used log data fields in audit log

Common Log Data Fields	Data Fields in auditd log
timestamp	msg=audit(TIMESTAMP)
source	proctitle, comm, exec, uid, gid, etc.
category	key
type	type
eventID	pid, ppid

4. Audit log records are automatically rotated daily and up to 14 achieved logs are kept at a time. When log rotates, the oldest archive will be deleted if 14 achieved logs exist
   - Audit log rotation rule can be modified in **`/etc/logrotate.d/auditd`**
5. The log timestamp is the local system time, which synchronizes with a remote Network Time Protocol (NTP) server
   - For time synchronization status and configuration, refers to timedatectl(1)

NOTE

Click the following link for more information on usages of auditd and log search
https://manpages.debian.org/bullseye/auditd/auditd.8.en.html
https://manpages.debian.org/bullseye/auditd/ausearch.8.en.html

Bootloader Audit Log

1. Log is stored in SPI flash with **1MB** storage size
2. Log can be viewed via **(3) Advance Setting > (6) View Bootloader Log** in Bootloader menu
3. Maximum number of logs is 4,000 records, where the oldest log will be overwritten when the maximum capacity is reached
4. The time stamp of the log read from the local Real-time Clock (RTC), which is synchronized with Network Time Protocol (NTP) server
5. Log format and log events are described below

Audit Log Structure

Header	Explanation	Possible Values
Time	Time stamp of the device	Format: [YYYY-MM-DDThh:mm:ss] For example: [2022-06-03T15:54:38]
User	Identifies the authenticated user	Admin
Category	Event category	• System. • Bootcfg (refers to boot configuration) • Install. • Security.
Event ID	ID of a logged event	1 ~ 15
Event Message	Description of the logged event	See below table for the list of events

Audit Events

Category	Event ID	Event Type	Event Message
System	1	Info	All bootloader configuration set to default
System	2	Info	Exit bootloader and reboot system
System	3	Info	Exit bootloader and boot to Linux
bootcfg	4	Info	Set boot configuration to default ok
bootcfg		Warning	Set boot configuration to default fail
bootcfg	5	Info	Set boot from SD/eMMC ok
bootcfg		Warning	Set boot from SD/eMMC fail
bootcfg	6	Warning	USB is not available on this device
bootcfg	7	Info	Bootarg and bootcmd changed
Install	8	Info	Install system image from TFTP ok
Install		Warning	Destination net unreachable
Install		Warning	Hash/Signature file not find
Install		Warning	System image file error
Install		Warning	File size is too large
Install		Warning	Upgrade system image fail
Install		Alert	System image authenticity check fail
Install	9	Info	Install system image from SD ok
Install		Warning	SD/eMMC device not find
Install		Warning	Hash/Signature file not find
Install		Warning	System image file error
Install		Warning	File size is too large
Install		Warning	Upgrade system image fail
Install		Alert	System image authenticity check fail
Secure		Warning	SD/eMMC device not find
Secure		Warning	Hash/Signature file not find
Secure		Warning	System image file error
Secure		Warning	File size is too large
Secure		Warning	Upgrade system image fail
Secure		Alert	System image authenticity check fail
Secure	11	Info	TFTP setting changed
Secure	12	Info	Login success
Secure		Warning	login fail
Secure	13	Alert	Boot failure due to system image integrity or authenticity check fail
Secure	14	Info	Admin password disabled

Category	Event ID	Event Type	Event Message
Secure		Info	Admin password enabled
Secure	15	Info	Admin password set to default
Secure	16	Info	Admin password changed
Secure	17	Info	Admin password policy changed
Secure	18	Info	Advance settings set to default
Secure	19	Info	Auto reboot threshold changed
Secure	20	Info	Login message changed
Secure	21	Info	Invalid Login Attempts changed
Secure	22	Info	Clear TPM ok
Secure		Warning	Clear TPM fail
audit	23	Info	View bootloader log ok

Audit Failure Response

The section is a guideline for protection of critical system functions in case of audit processing failure. Without an appropriate response to audit processing failure, an attacker's activities can go unnoticed, and evidence of whether the attack led to a breach can be inconclusive. Following are some common approaches:

1. **Log rotation**

   Log rotation is enabled by default in Moxa Arm-based computer to prevent audit storage capacity full. Refers to **Linux Audit Log** and **Bootloader Audit log** sections for details.

   In Linux, configure the logrotate to limit the disk space usage to prevent running out of space. The logrotate configuration file is at /etc/logrotate.config and all the files in /etc/logrotate.d/* to rotate the log file.

   This example we configure /etc/logrotate.d/rsyslog to rotate /var/log/syslog while it overs the size 2M with only 3 rotation.

```
/var/log/syslog

  {
    {
      rotate 3
      maxsize 2M
      ...
    }
  }
```

2. **Saving the logs in external storage**

   - For auditd, change the file path of parameter **log_file** in **/etc/audit/auditd.conf**
   - For rsyslog, change the default file path **/var/log/** in **/etc/rsyslog.conf** to external storage

3. **Use a centralized log Server**

   Use a centralized log managements system to collect and store the logs from Log from multiple devices. Refers to How to Set Up Centralized Logging on Linux with Rsyslog

4. **Assign appropriate action when audit storage space is full, or error occurs**

   You can configure **space_left** and **space_left_action** parameters in **/etc/audit/auditd.conf** to specify the remaining space (in megabytes or %) for low disk alert and what action to take. The actions are ignored, syslog, rotate, exec, suspend, single, and halt.

   In the example below, warning email will be sent to an email account specified in **action_mail_acct** parameter when the free space in the filesystem containing log files drops below 75 megabytes.

```
space_left = 75
space_left_action = email
```

   Configure **disk_full_action** and **disk_error_action** in **/etc/audit/auditd.conf** to specify what actions to take when the audit storage disk got error or is full. The actions are ignored, syslog, rotate (for disk full only), exec, suspend, single, and halt.

   Refers to auditd(8) for a detailed explanation of each action and parameters.

9. Programming Guide

Click the following link to download the ioThinx 4533 Programming Guide:

https://www.moxa.com/en/products/industrial-edge-connectivity/controllers-and-ios/advanced-controllersand-i-os/iothinx-4530-series#resources

The ioThinx 4533 Programming Guide includes the following sections:

Tutorials:

Shows users how to build code and use C or Python to access I/O data.

I/O Libraries:

Shows users how to access ioThinx 45MR, 45ML modules.

Module Information:

Shows users how to access module information.

Rotary Switch:

Shows users how to read the status of rotary switches.

User Defined LED Indicator:

Shows users how to access LED indicators.

Error Codes:

Provides the meaning of the return code to help users perform troubleshooting tasks.

A. Cycle Time Calculation

The controller's cycle time is defined as how much time the CPU needs to poll the status of all IO modules. This information is important since it allows users to make sure the controller can control their application within a designated time period. The cycle time calculation is based on the following table. A cycle time is calculated for each group of eight appended 45M modules. The cycle time of a group is the sum of the cycle time of the first module in the group (the times in column 1) plus the cycle times of the 2nd through 8th modules (the times in column 2) in the group. To calculate the cycle time of ioThinx 4533 Series CPU, simply add up the cycle times of all the groups connected to the ioThinx, and then round the time up to the nearest millisecond.

	Cycle time as 1st module in one group (µs)	Cycle time as 2nd to 8th module of the one group (µs)
45MR-1600	1200	100
45MR-1601	1200	100
45MR-2404	1300	100
45MR-2600	1200	100
45MR-2601	1200	100
45MR-2606	1200	100
45MR-3800	1300	200
45MR-3810	1300	200
45MR-6600	1500	300
45MR-6810	1500	300

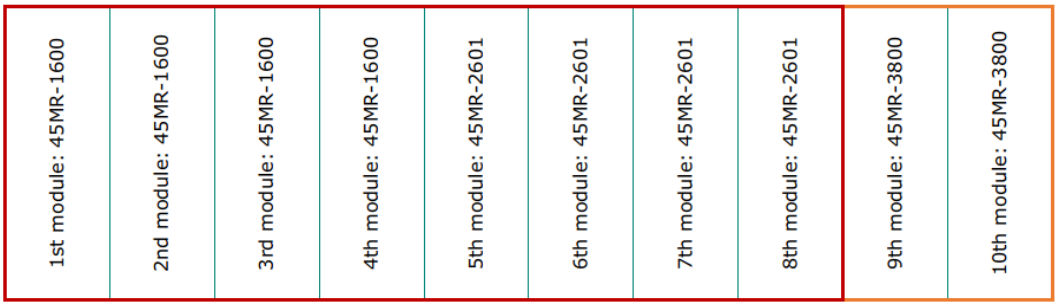
We provide two examples to illustrate cycle time calculations.

Case 1. 4-piece 45MR-1600 and 4-piece 45MR-2601

1st module: 45MR-1600	2nd module: 45MR-1600	3rd module: 45MR-1600	4th module: 45MR-1600	5th module: 45MR-2601	6th module: 45MR-2601	7th module: 45MR-2601	8th module: 45MR-2601

In this case, the eight modules form one group. The cycle time of this combination is 1900 µs = 1200 µs + 7 x 100 µs. The ioThinx 4533 Series will round up the cycle time to the ms level, and consequently the cycle time of this combination is 2 ms.

Case 2. 4 x 45MR-1600, 4 x 45MR-2601, 2 x 45MR-3800.



In this case, the 10 modules are separated in two groups. The first group is outlined in red above, whereas the second group is outlined in orange. The combination of modules in the first group is the same as in **Case 1**, which was shown to have a cycle time = 1900 µs. For the second group, the cycle time is 1500 µs = 1300 µs + 200 µs. Therefore, the total cycle time of the two groups is 3400 µs = 1900 µs + 1500 µs, which when rounded up to the nearest ms results in a total cycle time = 4 ms.