Getting Started with MPIC

Developer's Guide

TPD 20365 Rev 7
20 Oct 2022

Your worldwide
training partner
of choice

CAE

Contact Information

CAE Inc., 8585 Cote-de-Liesse, Montreal, Quebec, Canada, H4T 1G6

CAE Inc.
www.cae.com

Customer Services
caextranet.cae.com

Approval

Signature	Name	Title	Date	Revision

Revision History

Date	Comment	Name	Revision
29 Jan 2019	Draft	Patrick Gauvin	0
02 Feb 2021	Restricted shell procedure	Patrick Gauvin	1
04 Jun 2021	GAP assessment Part 1	Patrick Gauvin	2
19 Jul 2021	GAP assessment Part 2	Patrick Gauvin	3
08 Oct 2021	Improvement Drop	Patrick Gauvin	4
08 Nov 2021	Configure minimum password length	Patrick Gauvin	5
15 Dec 2021	SSH tunnel algorithm selection & authenticated log	Patrick Gauvin	6
20 Oct 2022	Front page rev and date fields changed	Patrick Gauvin	7

Contents

List of Figures

1 Introduction

The following guide will help developers understand the different components of the 5th interface generation (R5) by introducing the terminology and booting a MPIC for the first time.

1.1 Purpose

The purpose of this document is to help the developer boot a MPIC and use its operating system.

1.2 Scope

The developer will learn how to boot and communicate with a MPIC and important operating system details. The developer should be familiar with Linux and Windows. Finally, this document does not cover how to use a power supply.

1.3 Acronyms

Acronym	Definition
DB	Daughter Board
EMB	Embedded
MPIC	Multi-Purpose Interface Card
PCMIP	PCI Module Industry Pack
USB	Universal Serial Bus

2 Terminology

Many components are interacting with the Interface Manager module. This section provides definitions and descriptions of the terminology used throughout this guide.

2.1 Chassis

A chassis consists of the metal frame on which the circuit boards like interface cards are mounted. The chassis is designed to be placed in a cabinet.

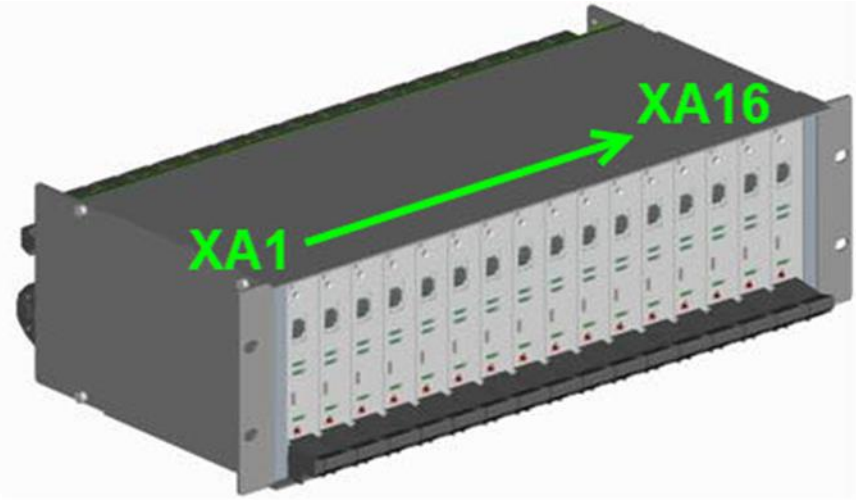

Figure 1 MPIC Chassis

2.2 Hardware Asset

A generic term to describe a package under configuration management which represents one or more interface cards, a cable, a hardware panel, and the software required to configure the interface card.

2.3 Interface Card

A peripheral device designed and configured to interact with hardware panels transmitting data into the simulation. Interface cards can be MPIC, MPIC-PCMIP or MPIC-EMB.

2.4 MPIC

The MPIC is a CAE interface card used on 7000XR series. The MPIC is intended to interface with one or more cockpit instruments/panels at the same time (referred as “one-

to-many" concept). The combination of MPIC, MPIC-PCMIP, MPIC-EMB, CAE DBs (Daughter Board) and PCMIP cards shall support any type of avionic instrument, real or simulated. The MPIC can support either 1 single or 1 double CAE DB card. The MPIC is designed to be slotted in a 3U chassis.

The MPIC has some similarities with the legacy USB-GPIM series, with the addition of more I/Os, more computing power and new functionalities such as local power supplies, PWM integral lighting, and AC power switching. The USB network is replaced with Ethernet.

Figure 2 MPIC Board

2.5 MPIC-PCMIP

The MPIC-PCMIP is a variant of the MPIC card. The main difference is that the MPIC DB slot has a custom type of interface (known as CAE DB) while the MPIC-PCMIP DB slot is PCMIP standard type. Typical use of the MPIC-PCMIP card is with a 429 or WXR PCMIP card to stimulate aircraft panels. The MPIC-PCMIP is designed to be slotted in a 3U chassis. The front plate has the same appearance as the MPIC and can be differentiated from the MPIC with its name labelled on top.

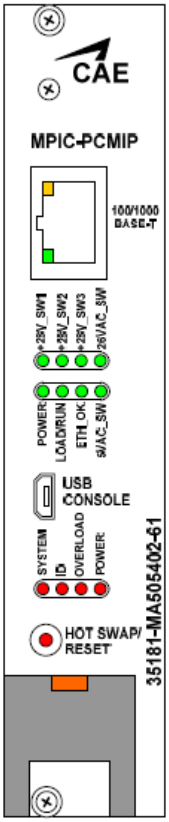

Figure 3 MPIC-PCMIP Front Plate

2.6 MPIC-EMB-CPU

The MPIC-EMB-CPU circuit card is based on the MPIC design. It is used on CAE equipment and is designed to be embedded, not mounted, in a chassis like the MPIC and MPIC-PCMIP cards. The upper primary card is always the MPIC-EMB-CPU and is usually mounted with one or many MPIC-EMB-IO cards based on what the CAE equipment I/O needs. The processor card can be used as a standalone card for additional processing.

An HDMI video connector is provided to directly drive digital instrument displays. The MPIC-EMB-CPU receives Ethernet communication and power and does not perform any I/O to the boards. The term MPIC-EMB is used to call an embedded standalone unit that contains an MPIC-EMB-IO boards.

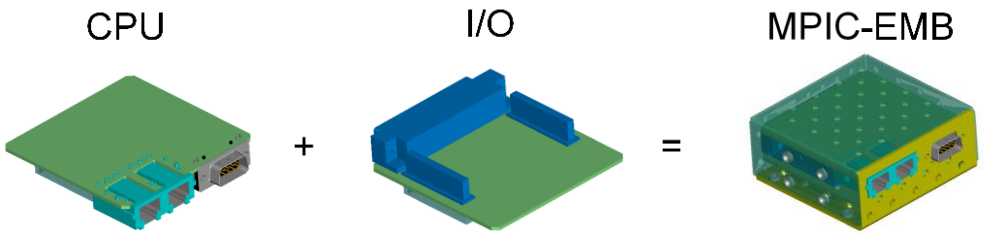

Figure 4 MPIC-EMB Assembly

2.7 MPIC-EMB-IO

The MPIC-EMB-IO is to be used on CAE equipment. The MPIC-EMB-IO card cannot be used standalone. It receives power from the MPIC-EMB-CPU high-density stack up connectors and provides I/O for the CAE instruments/panels and equipment.

The upper primary card is always the MPIC-EMB-CPU and controls the stack of I/O cards. Control logic and data are shared on the high-density stack up connectors. All MPIC-EMB-IOs are slaves and extend the stack upwards by means of bottom connectors.

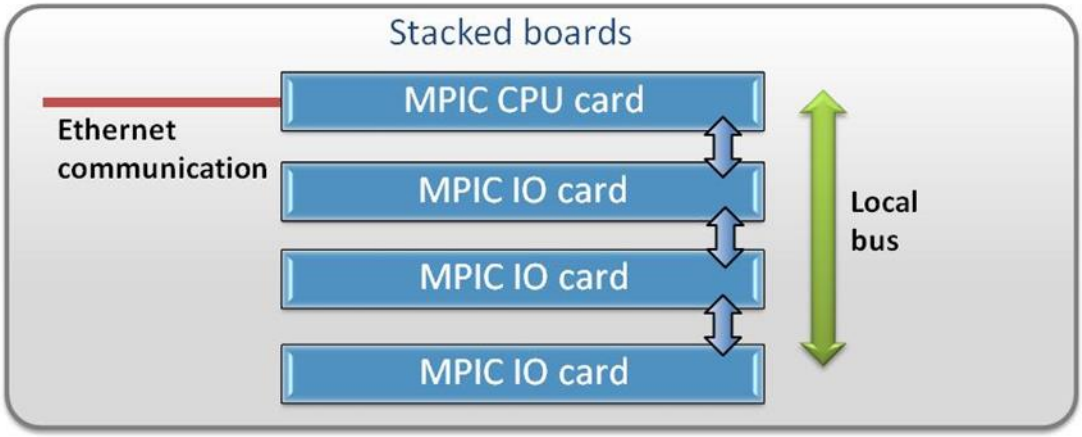

Figure 5 MPIC-EMB Block Diagram

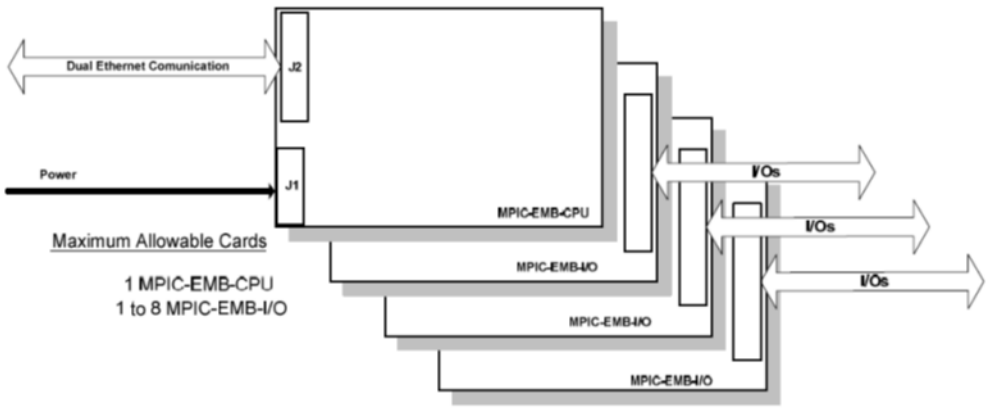

Figure 6 MPIC-EMB Overview

2.8 PCMIP

A PCMIP card is an extension card. It is a PCI-based mezzanine ANSI standard, suitable for small-footprint applications. This type of card can be mounted on a MPIC-PCMIP.

2.9 Daughter Board

A Daughter Board (DB) card is an extension card. This type of card can be mounted on a MPIC. All Daughter Boards used on 7000XR simulators are designed by CAE following a proprietary standard suitable for small-footprint applications.

3 Getting Started

3.1 Booting a MPIC or a MPIC-PCMIP

Insert the MPIC or the MPIC-PCMIP in the single-MPIC chassis and provide 28vdc to the single interface chassis using the DB9 power cord (see Figure 8). The MPIC or the MPIC-PCMIP will boot automatically once the power is applied.

Figure 7 Single-MPIC Chassis

3.2 Booting a MPIC-EMB-CPU

Use the DB9 connector with the black and red wires to connect the MPIC-EMB-CPU on a 28vdc power source. The MPIC-EMB-CPU will boot automatically once the power is applied.

Figure 8 DB9 Power Cord

3.3 Operating System

The operating system of any MPIC type is installed on a microSD card mounted on the board. It's a custom Linux distribution that support all the MPIC types.

3.4 Partitions

The microSD card contains 4 partitions: The kernel partition of the main operating system, the main operating system, a rescue operating system, and the application file system partition.

3.4.1 Kernel of the Main Operating System

The kernel of the main operating system is separated on its own partition. This partition is read-only.

3.4.2 Main Operating System

The main operating system partition contains the Linux file system used when the simulator is in operation. This partition is read-only.

3.4.3 Rescue Operating System

The rescue operating system partition contains a stripped Linux distribution for MPIC. U-Boot will boot on it after seven failed attempts to boot on the main partition. This partition is read-only. See the procedure section of this document to know how to change the boot partition.

3.4.4 The Application File System

The application file system partition is usually empty. It is used to keep development files without altering the main operating system partition. It is automatically mounted to "/home" by the main operating system.

3.5 Features

3.5.1 USB Console

A console can be accessed via a type B USB micro port on the front of the MPIC and the MPIC-PCMIP. The same USB port is available on the side of the MPIC-EMB-CPU (CCA1J3). **Do not use** the type B USB micro port located on the side of the MPIC and the MPIC-PCMIP and on the front of the MPIC-EMB-CPU.

Figure 9 USB Console Ports

To use the console, plug a USB cable (type B micro to type A) in the USB Console and in a computer. The first time you do so, Windows will install a driver. Every time you connect to a new MPIC, Windows should add a new USB Serial Port. You can see a list of the USB Serial Port under Ports (COM & LPT) in Device Manager. You can connect to this COM port with a serial terminal like PuTTY using a baud rate of 115200.

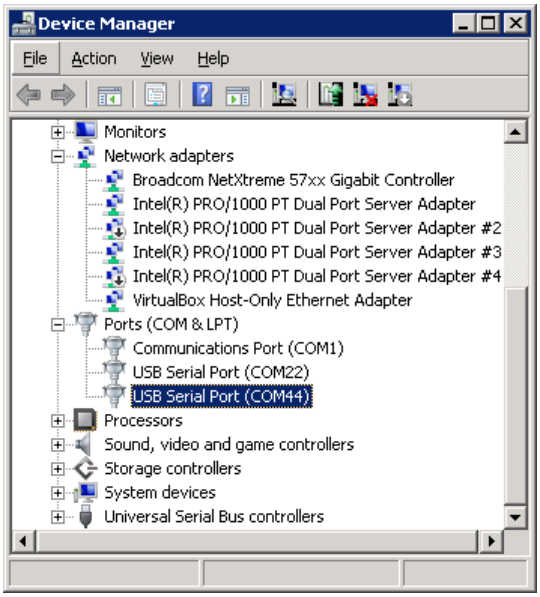

Figure 10 Device Manager

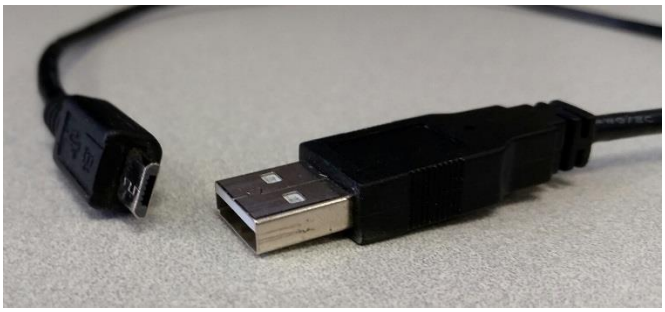

Figure 11 Type B Micro to Type A USB Cable

The only username is “admin”. The default password is “admin”.

3.5.2 Ethernet

3.5.2.1 MPIC and MPIC-PCMIP

The RJ45 port on the front of the MPIC and the MPIC-PCMIP is the ethernet input. A MPIC or a MPIC-PCMIP can share its network with up to three of its neighbors through the backplane of a chassis.

3.5.2.2 MPIC-EMB-CPU

The RJ45 port farther to the DB9 (GIGA) is the ethernet input. The other RJ45 port (FAST) is used to daisy chain the MPIC-EMB with another one.

Figure 12 MPIC-EMB Ethernet Cable

3.5.3 DHCP Client

By default, the operating system will try to acquire an IP address from a DHCP server. However, it is possible to use a static IP address instead. For, more information see the procedures section of this document to learn how to change it to a static IP address.

3.5.4 SSH Server

There is a SSH server on the standard SSH port 22 that gives you access to a remote shell and secure file transfer. The only username is “admin” and its default password is “admin”.

3.5.5 RAM disk

There is a RAM disk mounted in “/var/volatile”. Symbolic links are used on the main operating system partition to move files like logs on the RAM disk.

4 Procedures

4.1 Configure a Static IP Address

WARNING: this feature is not stable yet because another program reactivates the DHCP client later in the boot process. To uninstall this program, follow those steps.

1. Change the file system to be writeable

```
admin:~$ remount read-write
```

2. Remove the program

```
admin:~$ package uninstall OneInterfaceMpicPlatform
```

Follow those steps to configure a static IP address.

1. Connect to the MPIC using Serial USB with the user “admin”.
2. Run the following commands

```
admin:~$ network static <ip address>
```

Where <ip address> is a class C IP address.

```
admin:~$ network static <ip address> --netmask 255.255.0.0
```

Where <ip address> is a class B IP address.

```
admin:~$ network static <ip address> --gateway <gateway>
```

Where <ip address> is a class C IP address and;
Where <gateway> is the IP address of the gateway.

```
admin:~$ network dhcp
```

Configures the MPIC to get a dynamic IP address from a DCHP.

3. Reboot

```
admin:~$ power reboot
```

Examples

```
admin:~$ network static 192.168.0.2
admin:~$ network static 168.172.1.1 --netmask 255.255.0.0
admin:~$ network static 192.168.0.2 --gateway 192.168.0.1
admin:~$ network dhcp
```

4.2 How to update the image

1. Upload image files to the MPIC (might be already available on the MPIC)

```
admin:~$ scp <image_file> admin@<ip>:/home/admin
```

2. Connect to the MPIC using SSH or Serial USB with the user “admin”.

3. Reboot on the rescue partition.

```
admin:~$ power reboot --rescue
```

4. Reconnect to the MPIC using SSH or SERIAL USB. You should have a message “[RESCUE]” on the terminal.

5. Update the image. (Takes around 10 minutes)

```
admin:~$ update-image <image_file>
```

6. Once succeed, reboot in normal mode

```
admin:~$ power reboot --normal
```

4.2.1 Image files available on the TOE

4.2.1.1 cae-mx6qmpic-altered-content.tar.gz

The content of this file was altered, and the file wasn't re-signed.

Expected result:

```
admin:~$ update-image cae-mx6qmpic-altered-content.tar.gz
Cleaning environment data...[OK]
Extracting package...[OK]
Checking signature...[FAILED]
```

4.2.1.2 cae-mx6qmpic-wrong-signature.tar.gz

The content was re-signed with another private key.

Expected result:

```
admin:~$ update-image cae-mx6qmpic-wrong-signature.tar.gz
Cleaning environment data...[OK]
Extracting package...[OK]
Checking signature...[FAILED]
```

4.2.1.3 cae-*-mx6qmpic-3.0.*.0.tar.gz

Those are normal image file. Debug images have root available. Secure images are production versions. The version to evaluate is 3.0.37.0.

Expected result:

```
admin:~$ update-image cae-secure-mx6qmpic3.0.37.0.tar.gz
Cleaning environment data...[OK]
Extracting package...[OK]
Checking signature...[OK]
Checking environment...[OK]
Installing...[OK]
Updating the kernel...[OK]
```

4.2.1.4 rescuecae-debug-mx6qmpic-3.0.37.0.tar.gz

The rescue partition can be update in normal the same way the normal partition can be updated in rescue.

Expected result:

```
admin:~$ update-image rescuecae-debug-mx6qmpic-3.0.37.0.tar.gz
Cleaning environment data...[OK]
Extracting package...[OK]
Checking signature...[OK]
Checking environment...[OK]
Installing...[OK]
```

4.3 Filling the disk

1. Make sure the debug image is installed (see 4.2).
2. Log in with user: root password: caeadmin
3. Fill the disk

```
A200:~# dd if=/dev/zero of=/var/log/fill bs=1048576 count=10240
```

Note: Since the log are volatile, you'll have to do the command again if you reboot the TOE.

4.4 Changing the login banner

1. Connect to the MPIC using SSH or Serial USB with the user "admin".
2. Set the banner to a preset or custom text

```
admin:~$ banner set usdod
admin:~$ banner set-text $'---- Lightship Security ----\n\n'
```

4.5 Changing parameters

1. Connect to the MPIC using SSH or Serial USB with the user “admin”.
2. Change parameters
    - Deactivate the timeout in seconds, reset it to its default value or set it to a custom value

```
admin:~$ config set-timeout 0
admin:~$ config reset-timeout
admin:~$ config set-timeout 1200
```

    - Deactivate the number of authentication failures before the account is locked, reset it to its default value or set it to a custom value

```
admin:~$ config set-deny-count 0
admin:~$ config reset-deny-count
admin:~$ config set-deny-count 3
```

    - Set the time it takes after an authentication failure to unlock an account in seconds or reset it to its default value

```
admin:~$ config set-unlock-time 60
admin:~$ config reset-unlock-time
```

- Note: You can always see the current configuration

```
admin:~$ config print
```

- Note: The timeout parameter will be effective the next time you login.

4.6 Migrate from 3.0.17.0 to 3.0.37.0

1. Connect to the MPIC using SSH or Serial USB with the user “admin”.
2. Update the rescue partition with the migration file.

```
admin:~$ update-image /var/tmp/rescuecae-mx6qmpic-migration-
3.0.37.0.tar.gz
```

3. Without rebooting, log in root (no password), copy the configuration files and reboot.

```
root:~$ mount -o remount,rw /
root:~$ tar -C / -xf config.tar.gz
root:~$ mkdir /home/admin/.ssh
root:~$ chown admin:admin /home/admin/.ssh
root:~$ rm -rf /home/samba
root:~$ power reboot --rescue
```

4. Connect to the MPIC using SSH or Serial USB with the user “admin”.
5. Update the normal partition normally and reboot.

```
admin:~$ update-image /var/tmp/cae-mx6qmpic-3.0.37.0.tar.gz
admin:~$ power reboot --normal
```

4.7 Running the tests

1. Connect to the MPIC using SSH or Serial USB with the user “admin”.
2. Run all the tests, the power on self-test, the FIPS test suite, the FIPS power on self-test or the integrity tests

```
admin:~$ run-test all # Integrity + FIPS test suite
admin:~$ run-test power-on # Integrity + FIPS user test suite
admin:~$ run-test fips # FIPS test suite
admin:~$ run-test fips-post # FIPS user test suite
admin:~$ run-test integrity # RPM Integrity (see the note below)
```

Note: The integrity is checked for the following packages: openssl-fips, openssl, openssh, cae-commands, cae-commands-extended

4.8 Change password

1. Connect to the MPIC using SSH or Serial USB with the user “admin”.
2. Run the change password procedure

```
admin:~$ authentication change-password
Changing password for admin.
Current password:
New password:
Retype new password:
passwd: password updated successfully
```

4.9 Authenticate with a key

1. Generate a key pair on the client

```
user:~# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_rsa.
Your public key has been saved in /home/user/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:tC+Z+qB9jnCMoN1LcrIYO/k41Pe3Gu4bQ1eBD7lFeNs user@client
The key's randomart image is:
+---[RSA 2048]----+
|          =o     |
|         = o.    |
|        . *.o    |
|       . o.o E   |
|  o     .S.      |
| + + +. .+       |
|+.+ B =++ .      |
|+= * *.==o       |
|++o o =O*..      |
+----[SHA256]-----+
user:~# cat .ssh/id_rsa.pub
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDXIRTrirlL7VUJ6CZrZ0YbLr2AxA9D5u
5w8x+u+ZiqahMGfAnX9eZpI1mzECIcwBr9HdTRoFbDOTbBSzvicSyqD/pwqE3U
pvpM3BXR5PNbtUQTwA/t4Tzo3uNyPLg3ThvmD1vPABBpbwGsYw42MORzfJuI1Q
z4wXF6iIuiMLtjObKaLdf1qsZeTh8CuE50izD3H5ApCXAf4Z14XpmRHS5Goc2K
lHqatlyXL17usZs7ZmwOE9pPF50u1J44f3SLo4dQYf5uGZzQFKtmeA96sMuIj0
PrMn/we+M2GRddc8pcKFEbH9QKGG9WiNK1ZN4EEUbq1XHWjYxzr6YW3o/UXoQl
user@client
```

2. Connect to the MPIC using SSH or Serial USB with the user “admin”.

3. Add the authorized key on the MPIC

```
admin:~$ authentication add-key ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDXIRTrirlL7VUJ6CZrZ0YbLr2AxA9D
5u5w8x+u+ZiqahMGfAnX9eZpI1mzECIcwBr9HdTRoFbDOTbBSzvicSyqD/pw
qE3UpvpM3BXR5PNbtUQTwA/t4Tzo3uNyPLg3ThvmD1vPABBpbwGsYw42MORz
fJuI1Qz4wXF6iIuiMLtjObKaLdf1qsZeTh8CuE50izD3H5ApCXAf4Z14XpmR
HS5Goc2KlHqatlyXL17usZs7ZmwOE9pPF50u1J44f3SLo4dQYf5uGZzQFKtm
eA96sMuIj0PrMn/we+M2GRddc8pcKFEbH9QKGG9WiNK1ZN4EEUbq1XHWjYxz
r6YW3o/UXoQl user@client
```

4. To remove an authorized key:

```
admin:~$ authentication remove-key user@client
```

4.10 Create new SSH host-keys

1. Connect to the MPIC using SSH or Serial USB with the user “admin”.

```
admin:~$ ssh-config new-host-keys
```

4.11 Add SSH host to known hosts

1. Connect to the MPIC using SSH or Serial USB with the user “admin”.
2. Add a host to the know_hosts files.

```
admin:~$ ssh-config add-host 192.168.0.2 ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBCsE4BKcdlIsN7wre/
ah4VcMFir35AoBrfJZzFdCrhKqQwAy/s5qFmKYgCC1XlMpJ1rK7aoeJpAHlNlgl9yw/Kk=
```

3. To remove a known host:

```
admin:~$ ssh-config remove-host 192.168.0.105
```

4.12 Add/Remove NTP servers

1. Connect to the MPIC using SSH or Serial USB with the user “admin”.
2. Add a NTP server

```
admin:~$ ntp add ntp.ovh.net
```

3. Remove a NTP server

```
admin:~$ ntp remove ntp.ovh.net
```

4. Add a NTP server with a key

```
admin:~$ ntp add-key 11:c2b35394ff38a92709d65cc9de9b7b898101969d
admin:~$ ntp add 192.168.0.2:11
```

5. Remove a NTP server with its key

```
admin:~$ ntp remove 192.168.0.2
admin:~$ ntp remove-key 11
```

4.13 Configure rsyslog for remote logging

1. Connect to the MPIC using SSH or Serial USB with the user “admin”.
2. Enable rsyslog for remote logging in plain text:

```
admin:~$ remote-syslog enable 192.168.0.2
admin:~$ remote-syslog enable 192.168.0.2 --protocol udp
```

3. Enable rsyslog for remote logging through SSH tunnel:
    1. Generate a RSA key pair.

```
admin:~$ ssh-config generate-auth-key
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDqfXkaU6duv5jx3iXpfFrbqmEceyqQUoqyJ+B1Zu
g7ewoI9CfMGlP+wGxcyNNBdNKZGhTHNg9H9P2/rz8l2got8vXB/f5QHsBunKsIUcOTR+PC
Ifork5Q7vbLsBzvfUoVI+gkiop3hj4j5IleDPlTC/iqZ625kOT8XxuLVAS12OsLYq16VS0
Pia6V+kYZIkg/WFkhnjstrEczKkzmuVwajRT1wAce1YbC/4kkdJnywCiAMK91wSuqQEvYM
kYvhuo/KyhyTW3PwV+P+5Ahseb+09rmD/QYGeGkhagxGPjD1y+2EITrHXNT7xu91++aiLb
kD+St0ceuWEYhHK95EB6LN root@DNSL1XA16
```

Note: Multiple keys can be managed by using the switch “--name". The default name is “rsa”.

    2. Add the printed public key to ~/.ssh/authorized_keys of the syslog server.
    3. Configure the MPIC to connect to the syslog server through a SSH tunnel.

```
admin:~$ remote-syslog enable user@log -ssh
The authenticity of host 'log (192.168.0.2)' can't be established.
ECDSA key fingerprint is
SHA256:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'log' (ECDSA) to the list of known hosts.
```

4. Disable remote logging:

```
admin:~$ remote-syslog disable
```

5. Remove authentication key:

```
admin:~$ ssh-config remove-auth-key
```

4.14 Configure minimum password length

1. Connect to the MPIC using SSH or Serial USB with the user “admin”.
2. Change the password minimum length to 25

```
admin:~$ authentication change-password-minimum-length 25
```

3. Try to change the password minimum length to 10

```
admin:~$ authentication change-password-minimum-length 10
authentication change-password-minimum-length failed: password minimum
length cannot be set to lower value than 15
```

5 Change log 2021-07-21

5.1 Breaking changes

- Because of space constraints, we had to compress the “fs.tar”. To reflect this change, we renamed fs.tar and fs.tar.sign to fs.tar.gz and fs.tar.gz.sign. It’s still possible to migrate from an older firmware to a newer by creating a transfer rescue file. Just untar fs.tar.gz and fs.tar.gz.sign and rename before creating a new tar.gz file that you can use to do the migration. Once the rescue partition is migrated, just update the normal partition like before. Please follow the migration procedure 4.6.
- The root password for the development version is now caeadmin. Note that root is not accessible in the production version.
- The missing of a step in the update-image command makes it impossible to log in rescue after the transition. Follow procedure 4.6 to copy those files manually.

5.2 Audit

- Logs regarding authentications are available in /var/log/auth.log and will be transfer on the remote syslog.
- The command update-image will now log into syslog when it starts, complete and fail.

5.3 Commands

- New command “banner” to change the login banner
- New command “config” to change parameters
- New command “timezone” to set the timezone
- New command “run-test” to run the self-test (integrity and the FIPS test suite)
- New command “authentication” to change the admin password or configure an RSA authentication key
- New command “ssh-config” to generate new SSH host keys or managed know hosts.
- New command “remote-syslog” to configure rsyslog and a SSH tunnel for rsyslog.
- New command “ntp” to configure NTP servers and keys
- New command “collect-log” to create a log archive.
- New command “process” to list all the processes.
- New command “thermal” to get the MPIC temperatures

- Authorized Linux commands added: dmesg, netstat, ssh-keygen
- Authorized Linux commands “hostname” to be removed.

5.4 Configurations

- sshd will now issue a rekey after 1 hour
- sshd will now issue a rekey when any of the two channels reaches 512Mb

6 Change log 2021-10-08

6.1 General

- FIPS with ACVP (https://github.com/lightshipsec/openssl-fips-2.0.16-acvp)
- log rotation
- Audits removed of /var/log/syslog
- Audits continue to be sent to the syslog server
- Only 3 log files remain: /var/log/auth.log, /var/log/syslog, /var/audit/audit.log
- Commands have a new log format
- The USB console cannot be locked and will always unlock the user account
- CAAM RNG test at boot

6.2 New logs

- /var/log/syslog
    - Trusted channel initialization and failures
    - Host being added to or removed from known_hosts
    - Authentication keys being generated or removed
    - Configuration being set or reset
    - Password being changed
    - Authorized key being added or remove
    - Time being synchronized by ntpd
    - Session timeout
- /var/log/auth.log
    - New host keys being generated
    - Session termination from a timeout or a user request

6.4 Commands

- realtime-os: new command to swap the kernel for a realtime one (Xenomai)
- timezone: new feature to get the timezone from a custom service
- persistent-syslog: saves /var/log/syslog into /home/admin so it can survive a reboot
- location: gets the location (hostname) of the device
- platform-sate: prints simulation information
- core-dump: enables or disables the core dump
- ssh-config: manages authentication keys for SSH tunnels (see changed procedure 4.13)
- remote-syslog: uses the new key mechanism from ssh-config (see changed procedure 4.13)

7 Change log 2021-11-08

7.1 General

- Configurable minimum password length

8 Change log 2021-12-15

8.1 General

- Algorithms selection for the SSH tunnel fixed
- SSH tunnel authenticated log in /var/log/syslog