Intel® QuickAssist Technology Software for Free Berkeley Software Distribution* (FreeBSD*)

Release Notes - Software version

Package Version: QAT.B.3.10.0-00017

June 2021

Document Number: 621446-009

Contents

Tables

Revision History

Document Number	Revision Number	Description	Revision Date
621446	009	3.10.0 Release • Enable PKE processing on AE0 with RL/DU FW for CPM1.6/1.7 devices • Adds support for CPM1.72 devices • QAT Debuggability Black Box tool • USDM memory driver allocation performance improvement.	June 2021
621446	008	3.9.1 Alpha Release • FW change to enable processing AE0 on RL/DU FW for CPM1.6/1.7	March 2021
621446	007	Updated Section 3.1 Compiling the Driver with additional step	March 2021
621446	006	3.9.0 Product Release • Updated New features • Configurable instance feature • Single thread configuration option • Support for P5300 devices	February 2021
621446	005	3.8.0 Product Release • Updated New features • Device utilization v2 for CPM1.6/1.7 • Added known issues QATE-66213 • Updated known issue QATE-30360 • Added resolved issues QATE-63079 and QATE-40359	October 2020
621446	004	3.7.0 Product release	June 2020
621446	003	3.6.0 Product release	April 2020
621446	02	3.5.0 Product release	December 2019
621446	001	Initial release, 3.4.0 Product release	September 2019

§

1 Description of Release

This document describes extensions and deviations from the release functionality described in the Release Notes that support Intel® QuickAssist Technology (Intel® QAT).

This software release is intended for platforms that contain:

- Intel® C62x Chipset
- Intel Atom® C3000 processor product family
- Intel® QuickAssist Adapter 8960/ Intel® QuickAssist Adapter 8970 (formerly known as "Lewis Hill")
- Intel® Communications Chipset 8925 to 8955 Series
- Intel® Atom® P5300 processor product family

1.1 New Features Added with this Release

1.1.1 Performance Improvement

Enable Public Key Crypto processing on AE0 for Intel® Communications Chipset 8925 to 8955 Series, Intel® Atom® C3000 and Intel® C62x Chipset devices families on RL/DU firmware.

Note: `RateLimitingEnabled`flag support from configuration file was removed. DUv2 measurement available by default

1.1.2 Added Support for QAT 1.72 (Generation 2) Device ID Equal to 0x18EE for PF and 0x18EF for VF

Note: For QAT firmware authentication, IMR2 support should be enabled in BIOS. The option may be available under: EDKII Menu > Platform Configuration > Miscellaneous Configuration > Enable IMR2 Support

1.1.3 QAT Debuggability Tool

The QAT Debuggability tool was designed to add customer-usable debug solution that can gather data in order to help diagnose issues. It is intended to help customer to identify issue root-cause in relatively short time and avoid putting large effort into the debugging process.

The QAT library does not perform extensive checks or input data validation which can cause device hangs and other unexpected behavior. The root-cause of these issues are hard to identify without advanced debugging techniques. Using the tool, the customer is given enough information to allow them to find and fix defects caused by probable QAT API misuse. This should be achieved without QAT-specific technical knowledge required.

1.1.3.1 Quick Start Instructions

The section provides details on enabling the Debugging feature.

1.1.3.1.1 Compiling the Package

Step 1: Configure the driver with Debuggability feature enabled

```
# ./configure --enable-icp-qat-dbg
```

Step 2: Build and install driver

```
# gmake install
```

Configuration

QAT debuggability may be configured via dedicated section in QAT driver configuration file. Here is an example how this section can look like:

```
############################################
# QAT Debuggability Section
# Debug levels description:
#   0: no data collection
#   1: API calls data collection
#   2: FW calls data collection
#   3: combined level 1 and 2
############################################
[DEBUG]
Enabled = 1   # 0=collecting data disabled,1=collecting data enabled
DebugLevel = 2
NumBuffers = 128 #Number of buffers for data storage/device. [50-1000]
BufferSizeMB = 4  # Size of each buffer in MB.  [2-4]
LogDir = "/qat_crash" # Directory path for crash dumps
DumpOnProcessCrash = 0 # 0=Do not dump on crash,1=Dump on QAT crash
LogDirMaxSizeMB = 4096 # Maximum size of crash dump director. [1000+]
ContSyncEnabled = 0 # 0=no ongoing sync, 1=perform ongoing sync
ContSyncLogDir = "/qat_logs" # Path to directory for continuous sync
ContSyncMaxLogFiles = 10 # Max number of continuous sync files[10-100]
ContSyncMaxLogSizeMB = 100 #Max size of individual sync file[100-1000]
```

Note: Package is installed with debuggability section already added to configuration flies – but feature is disabled by default.

Step 3: Perform configuration restart

```
# adf_ctl restart
```

Step 4: Check debug configuration:

```
# qat_dbg_ctl status
QAT debuggabilty configuration:
    Device: 0
           Debug level: 2
           Buffer pool size: 128
           Buffer size in MB: 4
```

```
        Crash dump on client process: 0
        Synchronization mode: dump on crash
        Crash dump directory: /qat_crash
        Crash dump directory max size in MB: 4096
    QAT debuggability synchronization daemon running: Pid:
        76493
```

Note: If feature is enabled – "qat_dbg_sync_daemon" should be up and running. Daemon is initialized automatically by adf_ctl during loading configuration.

1.1.3.2 Collecting and Analyzing Data

The post-processing tool (`qat_dbg_report`) provides the following utilities:
Audits:

- Physical address used in FW request and sgls
- Return codes in FW responses
- Flat buffers and SGL buffers lengths based on cipher algorithm

Listings:

- Lists all collected entries sorted by sent/extraction time

Triggers:

- Manual trigger to dump content of debug buffers to configured location

1.1.3.3 Usage Example (With Continuous sync Disabled)

Pre-requisite: Verify that at least one device is configured with debug enabled and cont-sync feature disabled (ContSyncEnabled = 0)

Step 1. Generate payload

```
# ./build/cpa_sample_code signOfLife=1
```

Step 2. Trigger crash-dump manually to trigger data collection:

```
# qat_dbg_report command=dump dev=0
```

Step 3. Check collected data by using qat_dbg_report tool:

```
# qat_dbg_report path=/qat_crash/qat_crash_dev_<dev_id>_<timestamp>
command=list limit=0

=========================================================================
Building index…
DONE
    Overall indexed 309416 msgs.
        Requests: 154708 (Sym: 11280, PKE: 143232, DC: 196)
        Responses: 154708
        API calls: 0
```

Note: You can use different limit or audi commands instead of "list" to execute this test.

1.1.4 USDM Memory Allocation Performance Improvement

Reduced number of cycles required for memory allocation/deallocation.

1.2 Limitations with this Non-Production Release

- FreeBSD* as a host environment with Intel® QAT is not supported
- Any version of FreeBSD* other than v11.3 or v12.1 is not supported
- Mask Generation Function (MGF) and stateful compression are not supported starting with the 3.10 release for CPM 1.6, CPM 1.7x devices
- Symmetric session update feature is not supported
- Non-deterministic Random Bit Generator (NRBG) is not supported
- The HMAC-based Extract-and-Expand Key Derivation Function (HKDF) operational data has to be allocated with the Unified System Diagnostic Manager (USDM) to be pinned in physical memory
- No inline support
- SHA3 stateful is not currently supported

Note: There are known issues with this release of the driver, as described in Known-Issues within this Project.

1.3 Package Version

The following table shows the OS-specific package versions for each platform supported in this release.

Table 1. Package Version

Chipset or SoC	Package Version	SHA256 Checksum
Top-Level Package	QAT.B.3.10.0-00017.tar.gz	49390adfa1e85bebbffdd805f4d320cc2abad4d401323762263ec9e730501c4b

1.4 Licensing for FreeBSD* Acceleration Software

The acceleration software is provided under the following license, as listed in the table below.

Note: When using or redistributing dual-licensed components, you may do so under either license.

Table 2. Licensing for FreeBSD* Acceleration Software

Component	License	Directories
User Space Library	Berkeley Software Distribution (BSD)	./quickassist/build_system ./quickassist/include ./quickassist/lookaside ./quickassist/utilities/osal
Kernel space driver	BSD	./quickassist/qat/drivers ./quickassist/utilities/adf_ctl
User Space DMA-able Memory Driver	BSD	./quickassist/utilities/libusdm
Libcrypto	OpenSSL*	./quickassist/utilities/osal /src/linux/user_space/openssl
CPM Firmware	Redistribution	./quickassist/qat/fw
Calgary corpus and Canterbury corpus test files	Public domain	./quickassist/lookaside/access_layer /src/sample_code/performance/compression

1.5 Intel® QAT Application Program Interface (API) Updates

There are no Application Program Interface (API) changes in this release.

1.6 Technical Support

Intel® offers support for this software at the API level only, defined in the programmer's guide and API reference manuals listed in Section 2.1.1, Related Documents.

1.7 Environmental Assumptions

The following assumptions are made about the deployment environment:

- The driver object/executable file on disk should be protected using the normal file protection mechanisms so that it is writable only by trusted users, for example, a privileged user or an administrator.
- The public key firmware image on disk should be protected using normal file protection mechanisms so that it is writable only by trusted users, for example, a privileged user or an administrator.
- The Intel® QAT device should not be exposed (via SR-IOV) to untrusted guests.

- The Intel® QAT device should not be exposed (via the "user space direct" deployment model) to untrusted users.
- Dynamic random – access memory (DRAM) is considered to be inside the trust boundary. The standard memory-protection schemes provided by the Intel® architecture processor and memory controller, and by the operating system, prevent unauthorized access to these memory regions.
- Persistent keys were not considered, but the storage media are also found inside the cryptographic boundary. The driver exposed device file should be protected using the normal file protection mechanisms so that it could be opened and read/written only by trusted users.

§

2 Where to Find Current Software

This chapter provides a list of related documents and location of a list of files provided in this software release.

2.1 List of Files in Release

The Bill of Materials (BOM), sometimes referred to as the BOM, is included as a text file in the released software package. This text file is labeled a file list and is located at the top directory level for each release.

2.1.1 Related Documents

Table 3. Intel® QAT Related Documentation

Document Title	Reference Number
Intel® QuickAssist Technology API Programmer's Guide	330684
Intel® QuickAssist Technology Cryptographic API Reference Manual	330685
Intel® QuickAssist Technology Data Compression API Reference Manual	330686
Intel® QuickAssist Technology Performance Optimization Guide	330687
Using Intel® Virtualization Technology (Intel® VT) with Intel® QuickAssist Technology Application Note	330689
Intel® QuickAssist Technology Driver for FreeBSD*	https://01.org/intelquickassisttechnology

Note: Refer to https://01.org/intel-quickassist-technology for Intel® QAT program documentation.

2.2 Terminology

Table 4. Terminology

Term	Description
AEAD	Authenticated encryption with associated data
API	Application program interface

Term	Description
BOM	Bill of Materials
BSD	Berkeley Software Distribution
CNV	Compress and Verify
DRAM	Dynamic random –access memory
ESP	Enterprise Solution Platform program
FreeBSD*	Free Berkeley Software Distribution
GPL	General Public License
HKDF	HMAC-based Extract-and-Expand Key Derivation Function
Intel® QAT	Intel® QuickAssist Technology
IPsec	Internet Protocol Security
NRBG	Non-deterministic Random Bit Generator
MGF	Mask Generation Function
OS	Operating System
SADB	Security Association Database
SR-IOV	Single Root I/O Virtualization
PF	Physical Function
RAS	Remote Access Service
RDK	Reference Design Kit
RHEL*	Red Hat Enterprise Linux*
SOL	Sign-of-Life
UDP	User Datagram Protocol
USDM	Unified System Diagnostic Manager
VF	Virtual Function

§

3 Intel® QAT Driver Package Installation on FreeBSD* Environment

The user must have root privileges to perform the compiling of the drivers. Refer to Section 3.1 on how to compile the Intel® QAT Drivers.

3.1 Compiling the Driver

1. Copy package onto the system.
2. Extract package.

```
# cd /root/

# mkdir QAT

# cd QAT

# tar -xzomf <path_to>/ QAT.B.3.10.0-00017.tar.gz
```

3. Set network proxy (if required)

```
# export http_proxy http://<proxy_server>:<proxy_port>
```

4. Install dependencies:
5. gmake:

```
# pkg install gmake
```

6. Automake and autoconf:

```
# pkg install automake

# pkg install autoconf
```

7. bash:

```
# pkg install bash
```

8. pkg-config:

```
# pkg install pkgconf
```

9. yasm:

```
# pkg install yasm
```

10. Setup the environment to build driver.

```
# cd /root/QAT/

# ./configure
```

11. Build and install driver

```
# gmake install
```

3.2 Compiling and Execute Performance Sample Code

1. Build the application using the following:

```
# cd /root/QAT/

# gmake samples-install
```

2. Use this script to run the application:

```
#  cpa_sample_code signOfLife=1  <- sign of life tests

#  cpa_sample_code               <- full application run
```

3.3 Uninstalling Driver

1. Bring down the driver:

```
# adf_ctl down
```

2. Uninstall the driver:

```
# cd /root/QAT/

# gmake uninstall
```

3.4 Functional Sample Applications

Refer to Table 4, *Intel® QAT Technology API Programmer's Guide* for a copy of the functional sample applications included in the package.

These applications can be built using these steps:

```
# cd /root/QAT

# export= ICP_ROOT `pwd`

# export= ICP_OS freebsd

# export= WITH_CMDRV 1

# cd ./quickassist/lookaside/access_layer/src/sample_code/

# gmake func
```

The functional applications are built and placed in the `./functional/build` directory. Here is an example of how to run the functional sample applications.

```
# cd ./functional/build

# ./eddsa_sample
```

§

4 Intel® QAT Software - Known Issues and Resolved Issues

The following are errata Known-Issues, Resolved Issues, and Resolved Enhancements for Intel® QAT FreeBSD* (without v1.7) release.

4.1 Known-Issues within this Project

The following errata tables are known issues with the Intel® QAT FreeBSD* release.

4.1.1 QATE-68760 - DC - Concurrent compression or decompression requests can encounter false CPA_DC_WDOG_TIMER_ERR errors by Intel® QAT

Title	DC -Concurrent compression or decompression requests can encounter false CPA_DC_WDOG_TIMER_ERR errors by Intel® QAT
Reference #	QATE-68760
Description	If the CPA_DC_WDOG_TIMER_ERR error is encountered for a given compression request and there are concurrent compression or decompression requests running, the concurrent compression or decompression requests can encounter false CPA_DC_WDOG_TIMER_ERR errors being returned by Intel® QAT.
Implication	Concurrent compression or decompression requests can encounter false CPA_DC_WDOG_TIMER_ERR errors by Intel® QAT.
Resolution	There is no solution available yet, since FreeBSD* driver does not support DCSessionUpdate feature.
Affected OS	FreeBSD* 12.1
Driver/Module	CPM IA - Compression

4.1.2 QATE- 30931- Process Exit with Orphan Rings when spawning multiple processes

Title	Process exit with orphan rings when spawning multiple processes
Reference #	QATE- 30931
Description	If multiple processes start a user space service access layer (icp_sal_userStart) and they all exist together, the Syslog may show a message "Process <PID> <NAME> exit with orphan rings.
Implication	A kernel panic might happen at reboot if an application is using Intel® QAT.

Title	Process exit with orphan rings when spawning multiple processes
Resolution	The suggested workaround is to fork the process only after the previous child process runs icp_sal_userStartMultiProcess successfully.
Affected OS	FreeBSD*12.1
Driver/Module	CPM IA - Common

4.1.3 QATE-30360 - LBG and DNV device pass-through available only on guests with PCIe

Title	LBG and DNV device pass-through available only on guests with PCIe support
Reference #	QATE-30360
Description	LBG and DNV devices require PCIe support on guests for correct device initialization. Without PCIe support on guest FreeBSD* kernel recognizes passed through devices as PCI instead of PCIe and does not allow reading and writing PCI registers above 0xFF, while SOFTSTRAP_CSR_OFFSET, required for correct initialization of LBG and DNV devices in pass-through mode, is 0x2EC.
Implication	LBG and DNV device pass-through feature not available on guests without PCIe support.
Resolution	Guests must be configured with PCIe support for pass-through mode to work correctly with LBG and DNV devices.
Affected OS	FreeBSD* 12.1
Driver/Module	CPM IA – Common

4.1.4 QATE-39216 - Kasumi test duration issue

Title	Kasumi test duration issue
Reference #	QATE-39216
Description	Sample code benchmark tests included in the software package
Implication	The performance degradation when running the sample code can be observed in case the system runs the excessive number of threads.
Resolution	Avoid calling the `cpaCyInstanceGetInfo2` function if possible (i.e., by caching the info data) and try to tune the FreeBSD* scheduler.
Affected OS	FreeBSD*12.1
Driver/Module	CPM IA - Crypto

4.1.5 QATE-66213 - Symmetric Device Utilisation data incorrectly reported for Intel® Communications Chipset 8925 to 8955 Series devices

Title	Symmetric Device Utilisation data incorrectly reported for Intel® Communications Chipset 8925 to 8955 Series devices
Reference #	QATE-66213
Description	Symmetric Device Utilization data reporting for Intel® Communications Chipset 8925 to 8955 Series devices is incorrect, especially for larger packet sizes (16k, 32k), when the device reaches maximum throughput. The device utilization is under reported with these larger packet sizes.
Implication	Symmetric crypto device utilization is under reported for larger packet sizes as a result of PCIe bandwidth limitations.
Resolution	The resolution of this issue is not yet known.
Affected OS	FreeBSD* 12.1
Driver/Module	CPM IA – DU

4.1.6 QATE-66213 - QAT API submissions with bad addresses that trigger DMA to invalid or unmapped addresses can cause a platform hang

Title	QAT API submissions with bad addresses that trigger DMA to invalid or unmapped addresses can cause a platform hang
Reference #	QATE-73180
Description	This version of the QAT hardware does not perform request checking. It follows that a malicious application can submit requests that can bring down an entire QAT endpoint, which can impact other QAT jobs associated with the hardware. Furthermore, if any QAT API submission have bad addresses that would trigger DMA to invalid or unmapped addresses, these can induce a platform hang. This presents a risk to be managed by the host and guest operating systems and other system policies. The exposure can extend to other guest operating systems or applications outside of the typical access boundary of the malicious guest or application.
Implication	All guest operating systems or other applications using QAT must be trusted, and/or other steps must be taken to ensure that an untrusted application or guest cannot submit incorrectly formatted requests.
Resolution	There is no workaround available. However, system policies (including limiting certain operating system permissions) can help to mitigate this issue.
Affected OS	FreeBSD* 12.1
Driver/Module	CPM IA - Common

4.1.7 QATE-66213 - SRIOV - Concurrent VF bring-up may fail

Title	SRIOV - Concurrent VF bring-up may fail
Reference #	QATE-73515

Title	SRIOV - Concurrent VF bring-up may fail
Description	If QAT VFs are started concurrently, it is possible that one or more of these may not succeed.
Implication	Some interrupts may be ignored and the VF driver start should be retried.
Resolution	The resolution of this issue is not yet known.
Affected OS	FreeBSD* 12.1
Driver/Module	CPM IA - Common

4.2 Resolved Issues

4.2.1 QATE-39335 - Compression instances do not work on Virtual Machine with Linux* Host QAT driver without CnVnR support

Title	Compression instances do not work on Virtual Machine with Linux* Host QAT driver without CnVnR support
Reference #	QATE-39335
Description	FreeBSD* QAT VF driver does not get host capabilities - the `CnVnR` support is enabled by default.
Implication	The driver may fail to start compression instances on Virtual Machine with VF driver if no `CnVnR` support on Host QAT driver firmware.
Resolution	The issue is resolved in 3.7.0 release.
Affected OS	FreeBSD* 11.3
Driver/Module	CPM IA - Compression

4.2.2 QATE-41486 - Misleading message observed in `dmesg` on LBG device with `LimitDevAccess = 1` set in the configuration file.

Title	Misleading message observed in dmesg on LBG device with LimitDevAccess = 1 set in the configuration file
Reference #	QATE-41486
Description	When using `LimitDevAccess = 1` with more than one device in upstate, the "`qatX: failed to get NumberCyInstaces value from config!`" message could be

Title	Misleading message observed in dmesg on LBG device with LimitDevAccess = 1 set in the configuration file
	observed in `dmesg` for other devices than configured one. This message indicates only that for the other devices, the configuration was not found, which is expected.
Implication	This is an internal message only and should not be a threat as an error.
Resolution	The issue is resolved in 3.7.0 release.
Affected OS	FreeBSD* 11.3
Driver/Module	CPM IA - Common

4.2.3 QATE-33751 - GEN - Library, and driver do not support devices enumerated in a PCI domain different than 0

Title	GEN - Library and driver do not support devices enumerated in a PCI domain different than 0
Reference #	QATE-33751
Description	The userspace driver and the Intel® QAT library cannot handle devices enumerated in a domain different than 0.
Implication	It is not possible to use the software in systems where the device is enumerated with a PCI domain different than 0.
Resolution	The issue is resolved in 3.7.0 release.
Affected OS	FreeBSD* 11.3
Driver/Module	CPM IA - Common

4.2.4 QATE-59671 - Point Multiplication for Curve25519 and Curve448 not available on FreeBSD* guest machine

Title	Point Multiplication for Curve25519 and Curve448 not available on FreeBSD* guest machine
Reference #	QATE-59671
Description	The SR-IOV environment uses a Linux* driver on the host machine. At the time of the v3.6.0 FreeBSD* release, the EC Mont Edwards API is not yet supported on Linux* (in v4.8.0 release), which limits these elliptic curves operations to the FreeBSD* host.
Implication	Timeout observed on `EcEd` asymmetric crypto requests.

Title	Point Multiplication for Curve25519 and Curve448 not available on FreeBSD* guest machine
Resolution	The latest release of the Linux* Driver (4.9.0) includes support for these algorithms. Ensure Linux*driver version 4.9.0 or later is used to support these algorithms in a FreeBSD* Guest OS.
Affected OS	FreeBSD* 11.3
Driver/Module	CPM IA - Crypto

4.2.5 QATE-52976 - AlgChain and HKDF threads cannot use the same cy instance

Title	AlgChain and HKDF threads cannot use the same cy instance
Reference #	QATE-52976
Description	Possible bus error when symmetric and HKDF operation shares the same instance due to the request being overwritten.
Implication	It is impossible to share the same instance for symmetric and HKDF operations.
Resolution	The issue is resolved in a v3.6.0 release.
Affected OS	FreeBSD* 11.3
Driver/Module	CPM IA - Common

4.2.6 QATE-31888 - Possible performance degradation

Title	Possible performance degradation
Reference #	QATE-31888
Description	The integrated configuration for the FreeBSD* kernel is not optimized for all relevant Intel® QAT driver scenarios (issue with threading and scheduling).
Implication	Degradation of Intel® QAT data throughput can be observed in the deployment with FreeBSD*. The use cases: • sharing the same core for the threads using request ring (submission/working thread) and response ring (polling thread) • sharing the same core for among more working threads • an extensive number of threads waiting on mutex queue for responses
Cd /Resolution	The issue is resolved in a v3.6.0 release.

Title	Possible performance degradation
Affected OS	FreeBSD* 11.3
Driver/Module	CPM IA - Common

4.2.7 QATE-5092 - AES-XTS does not support buffers sizes that are not a multiple of 16B

Title	AES-XTS does not support buffers sizes that are not a multiple of 16B
Reference #	QATE-5092
Description	A single request with a data size that is not a multiple of 16B for AESXTS will fail with an invalid `param` check.
Implication	The user cannot submit AES-XTS Crypto requests with buffers that are not multiples of 16B
Resolution	The issue is resolved in the v3.6.0 release.
Affected OS	FreeBSD* 11.3
Driver/Module	CPM IA – Crypto

4.2.8 QATE-7325 - AES-GCM operation with zero-length plain text results in an incorrect tag result

Title	AES-GCM operation with zero-length plain text results in an incorrect tag result
Reference #	QATE-7325
Description	Sending an AES-GCM operation with zero-length plain text may generate an incorrect tag result
Implication	Potentially harmful record errors and failing connections
Resolution	The issue is resolved in the v3.6.0 release.
Affected OS	FreeBSD* 11.3
Driver/Module	CPM IA - Crypto

4.2.9 QATE-41846 - GEN – Intel® QAT API submissions with bad addresses that trigger DMA to invalid or unmapped addresses can cause a platform to hang

Title	GEN – Intel® QAT API submissions with bad addresses that trigger DMA to invalid or unmapped addresses can cause a platform to hang
Reference #	QATE-41846
Description	This version of the Intel® QAT hardware does not perform request checking. It follows that a malicious application can submit requests that can bring down an entire Intel® QAT endpoint, which can impact other Intel® QAT jobs associated with the hardware. Furthermore, if any Intel® QAT API submission have bad addresses that would trigger DMA to invalid or unmapped addresses, these can induce a platform hang. This presents a risk to be managed by the host and guest operating systems and other system policies. The exposure can extend to other guest operating systems or applications outside of the typical access boundary of the malicious guest or application.
Implication	All guest operating systems or other applications using Intel® QAT must be trusted, and/or other steps must be taken to ensure that an untrusted application or guest cannot submit incorrectly formatted requests.
Resolution	The issue is resolved in the v3.6.0 release.
Affected OS	FreeBSD* 11.3
Driver/Module	CPM IA - Crypto

4.2.10 QATE-41745 - Restore and Resize function in PKE code incorrectly freeing memory

Title	Segmentation fault when using inputs on QUAD word boundaries
Reference #	QATE-41745
Description	When using EC's `cpaCyEcPointMultiply` or `cpaCyEcPointVerify` with an aligned size of input parameters to four, eight, or nine `quadwords` (4 * 8B , 8 * 8B or 9 * 8B), a segmentation fault occurs.
Implication	Application crashes caused by a `segfault`.
Resolution	The issue is resolved in the v3.5.0 release.
Affected OS	FreeBSD* 11.2
Driver/Module	CPM IA - Common

4.2.11 QATE-40630 - Hang of asymmetric crypto engines might not be detected by heartbeat

Title	Hang of asymmetric crypto engines might not be detected by heartbeat
Reference #	QATE-40630
Description	Heartbeat might not detect a hang of an asymmetric crypto engine.
Implication	The device might be reported as responsive even if one of the engines is hung.
Resolution	The issue is resolved in 3.4.0 release.
Affected OS	FreeBSD* 11.2
Driver/Module	CPM IA - Common

4.2.12 QATE-40628 - Access to /dev/qat_adf_ctl allows a limited-trust user to reconfigure or reset the Intel® QAT endpoint

Title	Access to /dev/qat_adf_ctl allows a limited-trust user to reconfigure or reset the Intel® QAT endpoint.
Reference #	QATE-40628
Description	The device /dev/qat_adf_ctl provides a number of ioctls. Some ioctls are used by regular users of Intel® QAT for ring reservation and querying the configuration values. Others are used to reconfigure or reset the device. With the current implementation, any user that can use Intel® QAT for crypto or compression service can also reconfigure, bring down, or reset the device. These admin capabilities should be limited to admin users.
Implication	A user with access to /dev/qat_adf_ctl can reconfigure, bring down, or reset the device.
Resolution	The issue is resolved in the v3.4.0 release.
Affected OS	FreeBSD* 11.2
Driver/Module	CPM IA - Common

4.2.13 QATE-40627 - Destination buffer is overrun in a Digest Verify + Decrypt that does not reserve output space for the digest

Title	Destination buffer is overrun in a Digest Verify + Decrypt that does not reserve output space for the digest
Reference #	QATE-40627
Description	When the field `verifyDigest` in `CpaCySymSessionSetupData` is set to `CPA_TRUE`, the digest is written back to the destination buffer even if there is not allocated space in the destination buffer for it.
Implication	Unallocated memory can be overwritten
Resolution	The issue is resolved in the v3.4.0 release.
Affected OS	FreeBSD* 11.2
Driver/Module	CPM IA - Crypto

4.2.14 QATE-63079 - cpaDcResetSession may not wait until all flights are processed prior to clearing the inflight counters

Title	cpaDcResetSession may not wait until all flights are processed prior to clearing the inflight counters
Reference #	QATE-63079
Description	Excluding the compression session using the Data Plane API, cpaDcResetSession does not wait until all flights are processed prior to clearing the inflight counters. This is not correct behaviour since callback counters are reset before all the in-flight requests are processed.
Implication	If the session is reset while there are in-flight requests, segmentation faults and other unexpected application behaviour may be encountered.
Resolution	The issue is resolved in 3.8.0 release.
Affected OS	FreeBSD* v12.1
Driver/Module	CPM IA - Common

4.2.15 QATE-40359 - Multiprocess 32 with LimitDevAccess = 0 fails with OpenSSL* Speed tests

Title	Multiprocess failure with NumProcesses > 16 for LBG/DNV and NumProcesses > 32 for CLC and LimitDevAccess = 0
Reference #	QATE-40359
Description	The `multiprocess` application that uses more than 16 processes for LBG/DNV and 32 processes for CLC fails during bundle allocation.
Implication	It is impossible to successfully run the `multiprocess` application with more processes than 16 for LBG/DNV and 32 for CLC.
Resolution	The issue is resolved in 3.8.0 release.
Affected OS	FreeBSD* 11.3
Driver/Module	CPM IA - `Multiprocess`

§