Barracuda KTINA FIPS Crypto Module

Version 7.1

FIPS 140-2 Non-Proprietary Security Policy

Level 1 Validation

Document Version 1.9

Prepared By:

ICSAlabs
An Independent Division of Verizon Business

Revision History

Version	Modification Date	Modified By	Description of Changes
1.0	2016-09-26	ICSA Labs	Initial Document
1.1	2016-10-17	ICSA Labs	Miscellaneous typos/spacing; Include module version number in each section where the module is initially referenced Re-worded paragraph 2.1 to describe automatic execution of integrity and self tests Table 2-3: changed heading to "Key Lengths"; added "Integrity Check" to HMAC Use; added 168 to Triple-DES key length. Section 4, clarified the password length and valid values. Table 4-1: clarified that self tests are re-run Section 9: include ref to IG9.10; split out individual KATs; Re-worded paragraph 3
1.2	2016-10-20	ICSA Labs	Removed AES 192
1.3	2016-11-09	ICSA Labs	Added Triple-DES weak key detection and non-random AES key detection to "Mitigation of Other Attacks"
1.4	2016-11-29	ICSA Labs	Added CAVS certification numbers; Included instructions for disabling weak key checks; corrected table 4-1 for HMAC integrity tests and SHA calculation
1.5	2016-12-05	ICSA Labs	Updated to reflect individual SHA KATs
1.6	2016-12-14	ICSA Labs	Added User role to Zeroize/Uniinitialize service
1.7	2016-12-23	ICSA Labs	Added ktina_meth_first, ktina_meth_next and ktina_meth_hash_digest_size to the services table (4-1); changed services entry from "initialization for the use.." to "prepare for the use…"
1.8	2017-03-21	ICSA Labs	Modifed sections 1.1, 2.3, 7,3, 7.5 and tables 2-3 and 4-1 to comply with CMVP review comments.
1.9	2017-03-29	ICSA Labs	Modified Tables 2-3 and 4-1 and opening paragraphs in section 4 to comply with CMVP review comments

Table of Contents

Table of Figures

Table of Tables

1 Introduction

1.1 Purpose

This is a non-proprietary Cryptographic Module Security Policy for the Barracuda KTINA FIPS Crypto Module 7.1 from Barracuda Networks. It provides detailed information relating to the Federal Information Processing Standard (FIPS) 140-2 security requirements for conformance to security Level 1, and instructions on how to run the module in a secure FIPS 140-2 approved mode.

2 Cryptographic Module Specification

The Barracuda KTINA FIPS Crypto Module 7.1 is a software kernel module that provides FIPS 140-2 approved cryptographic functions for other kernel modules in Barracuda security products that use Barracuda NextGen Firewall and Control Center OS 7. The FIPS 140-2 validation of the Barracuda KTINA FIPS Crypto Module is comprised of the file *ktinafips.ko*.

2.1 Module Overview

The Barracuda KTINA FIPS Crypto Module 7.1 is a software-based cryptographic module. Table 2-1 provides a list of platforms, operating systems and processors on which the Barracuda KTINA FIPS Crypto Module was tested.

Hardware Test Platforms	Operating System	Processor	Processor Optimization
Dell PowerEdge R320	*Barracuda NextGen Firewall and Control Center OS 7 on Microsoft Windows 2012 (64-bit) Hyper-V*	*Intel Xeon*	*None*
Dell PowerEdge R320	*Barracuda NextGen Firewall and Control Center OS 7 on Microsoft Windows 2012 (64-bit) Hyper-V*	*Intel Xeon*	*AES-NI*

Table 2-1: Tested Configurations

The logical cryptographic boundary of the module is the kernel module “Barracuda KTINA FIPS Crypto Module” (ktinafips.ko). It is contained in the physical boundary of the general purpose computer (GPC) on which the module resides. Figure 2-1 describes the GPC physical boundary, the virtual machine, the Barracuda KTINA FIPS Crypto Module logical boundary, and the relationships among them.

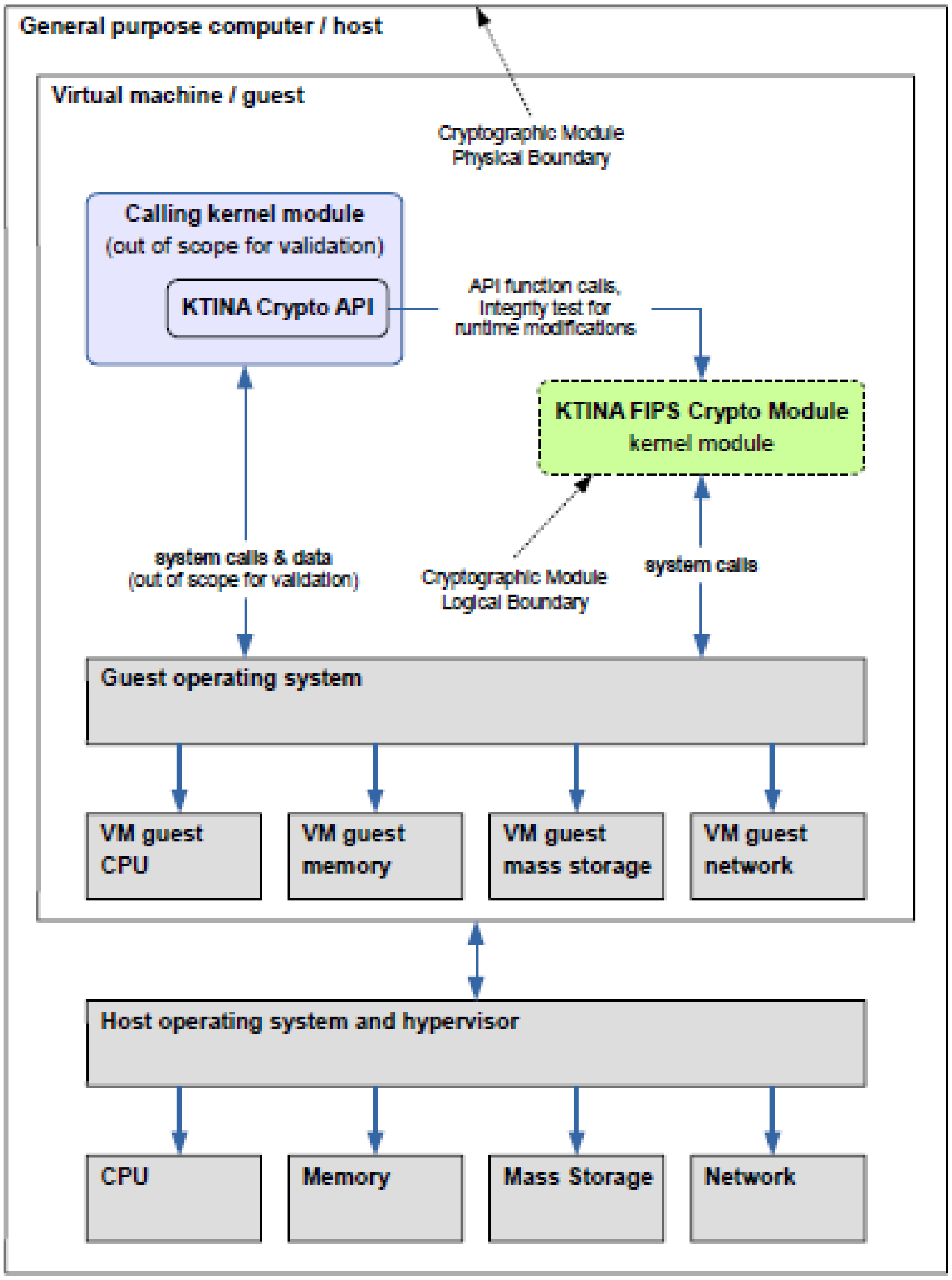
General purpose computer / host
Virtual machine / guest
Cryptographic Module Physical Boundary
Calling kernel module
(out of scope for validation)
KTINA Crypto API
API function calls, Integrity test for runtime modifications
KTINA FIPS Crypto Module
kernel module
system calls & data
(out of scope for validation)
Cryptographic Module Logical Boundary
system calls
Guest operating system
VM guest CPU
VM guest memory
VM guest mass storage
VM guest network
Host operating system and hypervisor
CPU
Memory
Mass Storage
Network

Figure 2-1: Logical Block Diagram

2.2 Security Levels

Per FIPS 140-2 terminology, the Barracuda KTINA FIPS Crypto Module 7.1 is a multi-chip standalone module that meets overall level 1 FIPS 140-2 requirements. Table 2-2 lists the validation levels for each section of the Barracuda KTINA FIPS Crypto Module:

Section	Section Title	Level
1	Cryptographic Module Specification	1
2	Cryptographic Module Ports and Interfaces	1
3	Roles, Services, and Authentication	2
4	Finite State Model	1
5	Physical Security	N/A
6	Operational Environment	1
7	Cryptographic Key Management	1
8	EMI/EMC	1
9	Self-tests	1
10	Design Assurance	3
11	Mitigation of Other Attacks	1

Table 2-2: Security Level per FIPS 140-2

2.3 Modes of Operation

The Barracuda KTINA FIPS Crypto Module 7.1 has only a FIPS Approved mode of operation. It is loaded automatically via the linux *insmod* command when the host operating system is booted. Upon loading, the module automatically performs integrity testing and algorithm self tests meeting the requirements outlined in *"Implementation Guidance for FIPS 140-2 and the Cryptographic Module Validation Program"* section 9.10. To use the module, a client component calls ktina_api_use_fips(), which re-runs the algorithm self tests and exposes the module's cryptographic function entry points to the caller. Once initialized, the Barracuda KTINA FIPS Crypto Module supports the FIPS Approved Algorithms listed in Table 2-3.

In order to support validation testing, the module supports an option to suppress weak key checking while in FIPS Approved mode of operation. To enable this option, the parameter "ktinaAllowWeak=1" must be supplied in the *insmod* command and the "allowWeakKeys" parameter of the ktina_api_use_fips function must be set to 1.

CAVP Cert	Algorithm	Standard	Modes	Key Lengths	Use
4150	AES	FIPS 197 SP 800-38A	CBC	128, 256	Data Encryption/Decryption
2720	HMAC	FIPS 198-1	SHA-1/256/512	KS<BS, KS=BS, KS>BS	Message Authentication and Integrity Check
3416	SHS	FIPS 180-4	SHA-1/256/512		Message Digest – Used only to support HMAC
2267	Triple-DES	SP 800-67	CBC	192 (168 independent bits, 112 bits of security)	Data Encryption/Decryption

Table 2-3: FIPS Approved Algorithms

3 Module Ports and Interfaces

The physical ports of the module include those of the GPC on which the module is executed, but are outside the scope of the FIPS 140-2 validation. The logical interface consists of an application program interface (API) through which consumers of the module's services may exact control, request status, or pass data in/out. The FIPS 140-2 interfaces are described in Table 3-1: FIPS 140-2 Logical Interfaces. The Barracuda KTINA FIPS Crypto Module API documentation includes all the inputs, outputs, control, and status parameters.

FIPS 140-2 Logical Interface	Implementation
Data Input	API with stack and register input parameters
Data Output	API with stack and register output parameters
Control Input	API with stack and register control parameters
Status Output	API with stack and register status parameters
Power Interface	N/A

Table 3-1: FIPS 140-2 Logical Interfaces

4 Roles, Services, and Authentication

The Barracuda KTINA FIPS Crypto Module 7.1 operates only in FIPS Approved mode and supports operators in either a Crypto-Officer (CO) role or User role. To initialize the cryptographic module, the CO supplies a factory-defined password to the ktina_api_start_fips() function. Once intialized, a user may invoke the ktina_api_use_fips() function with a factory-defined password in order to use the FIPS Approved functions. The module interface prohibits multiple users from accessing the module's services simultaneously. The pre-defined passwords set at module build time are 33 characters in length and may include any byte values. Passwords are validated by comparing an HMAC hash of the user-supplied value with an internally stored HMAC hash of the module's pre-defined password. The probability of a single random successful authentication attempt is $2^{-(8*33)}$. The module limits multiple authentication attempts to ten attempts per second. The probability of a successful random attempt or a false acceptance occurring in a 1 minute period is 600 x $2^{-(8*33)}$ which is less than one in 100,000. If authentication fails, the module remains in the current state and no cryptographic functions are accessible.

The module doesn't generate keys nor does it output any keys to the operator. Therefore, operator access to CSPs can never be R (read access). Write access is marked with W in the table below. The zeroization of keys is marked with Z. Services that make use of keys or do checks against the built-in crypto officer or crypto user password HMACs are marked with E for execute access. Status functions do not work with any keys or passwords, therefore they are marked with N/A.

The module provides the services listed in Table 4-1.

Service	Standard	Roles	Description	CSPs & Public Keys	API	**Key/CSP Access for Operator** W=Write R=Read E=Execute Z=Zeroize
AES-128/256 Encrypt/Decrypt (CBC Mode Only)	FIPS 197 SP 800-38A	User	Symmetric Encryption/ Decryption using the AES encryption Standard	AES Encrypt/Decrypt Key	ktina_meth_get_inst_from_name() ktina_meth_setkey() ktina_meth_encrypt()[1] ktina_meth_decrypt()[1]	W, E
Triple-DES Encrypt/Decrypt (CBC Mode Only)	SP 800-67	User	Symmetric Encryption using the Triple-DES encryption Standard	Triple-DES Keys Three-key: K1 != K2 != K3 != K1	ktina_meth_get_inst_from_name() ktina_meth_setkey() ktina_meth_encrypt()[1] ktina_meth_decrypt()[1]	W, E
HMAC-SHA-1/256/512	FIPS 198-1	User	Generate HMAC-SHA	HMAC Key	ktina_meth_get_inst_from_name() ktina_meth_hash_hmac_setkey() ktina_meth_hash_hmac_calc()	W, E
HMAC-SHA-256 for integrity check	FIPS 198-1	CO	Generate HMAC-SHA256	HMAC Key	memcheck_SHA256_hmac_calc()	E
SHA-1/256/512	FIPS 180-4	User	Generate Cryptographic Hash		ktina_meth_get_inst_from_name() ktina_meth_hash_init() ktina_meth_hash_update() ktina_meth_hash_final()	N/A
Initialization & Operator Authorization		CO	Prepare the module for FIPS approved use and re-run self-tests	Factory assigned password for CO role authentication	insmod ktinafips.ko ktina_api_start_fips()	W, E

[1] The ktina_meth_get_inst_from_name() function must be called prior to either the ktina_meth_encrypt() or ktina_meth_decrypt() functions to obtain a context pointer that is subsequently passed to the ktina_meth_encrypt/ktina_meth_decrypt functions. This defines the algorithm used for the operation (AES or Triple-DES).

Service	Standard	Roles	Description	CSPs & Public Keys	API	Key/CSP Access for Operator W=Write R=Read E=Execute Z=Zeroize
Prepare for the use of crypto functions and crypto user authorization		User	Prepare the module for the use of FIPS approved functions for the "User" role	Factory assigned password for User role authentication	ktina_api_use_fips()	W, E

Status / Version		User/CO	Retrieve the current status of the module or version information	None	ktina_meth_is_fips_mode()
					ktina_meth_get_state()
					ktina_meth_get_api_ready()
					ktina_meth_cipher_iv_size()
					ktina_meth_cipher_minkey()
					ktina_meth_cipher_maxkey()
					ktina_meth_cipher_defkey()
					ktina_meth_name()
					ktina_meth_type()
					ktina_meth_info()
					ktina_get_crypto_module_version
					ktina_meth_first()
					ktina_meth_next()
					ktina_meth_hash_digest_size()
						N/A

Zeroize/Uninitialize		User/CO	Zeroize the CSP's of an instance. This does not zeroize the factory-assigned passwords.	AES Key; Triple-DES Key; HMAC Key, Factory assigned password for CO or User role authentication	ktina_api_unuse_fips()
					ktina_api_stop_fips()
					ktina_meth_free_inst()
					(the first two will call ktina_meth_free_inst() internally to clear the method instances including any keys)
						Z, E (Zeroize keys)

Self-Test		CO	Performs integrity test (using HMAC-SHA256) and algorithm self-tests. These are always peformed at power-on and may optionally be run on –demand.	Factory assigned password for CO role authentication (not required for retest)	ktina_api_start_fips()
					ktina_api_retest_fips()
						W, E

Table 4-1: FIPS Approved Services with Roles/CSPs

5 Physical Security

The physical security requirements do not apply to the Barracuda KTINA FIPS Crypto Module 7.1 because the module is a FIPS 140-2 Level 1 software module and the physical security is provided by the host platform.

6 Operational Environment

The module operates on a General Purpose Computer (GPC) which is a modifiable operating system. The module was tested on the platforms defined in Table 2-1.

Being a kernel module that runs within the Barracuda NextGen Firewall and Control Center OS 7, ktinafips.ko supplies cryptographic services to other kernel modules only. It is through policy that only a single user may access these services at any particular time. In order to enforce this policy, the module is accessible only though a provided interface which prevents multiple users.

Like all loadable kernel modules, it is an extension of the the base kernel of the operating system and is protected as such.

7 Cryptographic Key Management

7.1 Critical Security Parameters (CSPs)

Table 7-1 contains a list of keys/CSPs used in the module. Sections 7.2-7.4 describe the generation, entry, storage, output and zeroization of the keys/CSPs used in the module.

CSP	Description
AES EDK	AES Encrypt/Decrypt Key
Triple-DES Symmetric Keys	Triple-DES Keys Three-key: K1 != K2 != K3 != K1
HMAC Key	Message Authentication Code Key
CO Auth Digest	Digest for Crypto Officer authentication
User Auth Digest	Digest for User authentication
Integrity HMAC Digest	Digest for detection of module corruption

Table 7-1: Module CSPs

7.2 Key Generation

The module does not support key generation.

7.3 Key Entry, Storage, Output

No keys are persisted by the module beyond the lifetime of the API calls. All keys/keying materials are entered into the module from the consuming application (i.e. “operator”) as plaintext parameters in RAM to the API functions. Keys/keying material originates within the physical boundary of the module by the **Barracuda Cryptographic Software Module** (CMVP certification #2458). No keys/keying material are output from the logical boundary of the module.

7.4 Zeroization

Temporarily stored keys/ keying material are zeroized when the CO calls the ktina_api_stop_fips function, the user calls the ktina_api_unuse_fips function, or when the module is unloaded via rmmod. The operating system protects system memory and process space from access by unauthorized users. The operator of the cryptographic module should follow the steps outlined in the documentation to ensure sensitive data is protected by zeroizing the data from memory when it is no longer needed.

7.5 Entropy

The module contains neither entropy source nor random number generators (RNG, DRBG, etc.). When keys or random numbers are required, they must be generated by the **Barracuda Cryptographic Software Module** (CMVP certification #2458) within the physical boundary of the GPC on which the module resides. Using the **Barracuda Cryptographic Software Module** assures that security strength requirements are met.

8 EMI/EMC

The module is a software module and was tested on standard GPC platforms that meet the applicable Federal Communication Commission (FCC) Electromagnetic Interference (EMI) and Electromagnetic Compatibility (EMC) requirements for business use as defined in Subpart B of FCC Part 15.

9 Self-Tests

The Barracuda KTINA FIPS Crypto Module 7.1 performs the required suite of self-tests upon initialization of the module. The self-tests are performed automatically without operator intervention and meet the requirements outlined in *"Implementation Guidance for FIPS 140-2 and the Cryptographic Module Validation Program"* section 9.10. The following self-tests are performed:

Self Tests:

- Software integrity HMAC SHA-256
- HMAC SHA-1 Known Answer Test (KAT)
- HMAC SHA-256 KAT
- HMAC SHA-512 KAT
- SHA-1 KAT
- SHA-256 KAT
- SHA-512 KAT
- 3-Key Triple-DES Encrypt KAT: CBC mode
- 3-Key Triple-DES Decrypt KAT: CBC mode
- AES 128 Encrypt KAT: CBC mode
- AES 128 Decrypt KAT: CBC mode
- AES 256 Encrypt KAT: CBC mode
- AES 256 Decrypt KAT: CBC mode

Conditional tests are not implemented as the module does not support any algorithm requiring such tests.

The module will enter an error state on failure of any self-test and prevent servicing of any subsequent requests for cryptographic functions. The module must be power cycled to remove it from the error state. Once power cycled, the power-on self-tests will be run. If all tests pass the module will move into an operational state. If any of the self-test fails the module will move back to the error state.

The self-tests can be performed on demand by the operator by invoking the ktina_api_retest_fips() function.

10 Design Assurance

Barracuda uses Subversion (SVN) for configuration/change management of source code. Documentation versioning is managed with Atlassian Confluence. All module source code and documentation is maintained on servers that are internal to Barracuda.

The Barracuda KTINA FIPS Crypto Module 7.1 is for use inside of Barracuda products. The module is a binary object module and is only distributed to the Barracuda development team as the FIPS 140-2 validated *ktinafips.ko* binary object. The module code has a computed HMAC SHA-256 embedded in it for the software integrity test. If there are any changes to the module or the HMAC SHA-256 the software integrity test will fail. The Barracuda development teams work in secure environments with controlled access. The module and the host application are installed on one of the operational environments listed in Table 2-1.

11 Mitigation of Other Attacks

By default this module protects against Triple-DES weak keys, semi-weak keys and possibly weak keys as described in SP 800-67 section 3.4.2. It also protects against non-random keys (all bytes of key being equal) for AES. This feature may be disabled for testing by following the instructions outlined in section 2.3

12 Crypto-Officer and User Guidance

For normal operation, the ktinaAllowWeak option must be disabled for the module when it is instantiated (which is the default setting if the ktinaAllowWeak option is omitted), and the allowWeakKeys parameter of the ktina_api_start_fips function must be set to 0. If the allowWeakKeys parameter differs from the allowWeakOption, the ktina_api_start_fips function will fail.

The calling application is the operator (crypto-officer or user depending on the password supplied) of the module. The Barracuda KTINA FIPS Crypto Module 7.1 is for use on a GPC. It is the responsibility of the calling application to secure any keys or CSPs which are stored outside of the logical boundary of the module. The module does not provide any persistent storage of keys or CSPs.

The module maintains internal context information to identify the current user. Attempts by other users (applications) to perform any of the module's functions result in the request being rejected.

When using the module, it is also the responsibility of the calling application to use a FIPS validated module to generate random values used for keys or other CSPs that are supplied to the Barracuda KTINA FIPS Crypto Module.

13 Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
API	Application Program Interface
CBC	Cipher Block Chaining
CO	Cryptographic Officer
CSP	Critical Security Parameter
DES	Data Encryption Scheme
EDK	Encrypt, Decrypt Key
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FCC	Federal Communications Commission
FIPS	Federal Information Processing Standard
GPC	General Purpose Computer
HMAC	Keyed-Hash Message Authentication Code
KAT	Known Answer Test
OS	Operating System
RAM	Random Access Memory
SHA	Secure Hash Algorithm
Triple-DES	Triple-DES