FIPS 140-2 Non-Proprietary Security Policy

Acme Packet VME

FIPS 140-2 Level 1 Validation

Software Version: S-Cz9.0

Date: January 24, 2024

Title: Acme Packet VME (v9.0) Non-Proprietary Security Policy
Date: January 24, 2024
Author: Acumen Security, LLC.
Contributing Authors:
Oracle Communications Engineering
Oracle Security Evaluations – Global Product Security

Oracle Corporation
World Headquarters
2300 Oracle Way
Austin, TX 78741
U.S.A.
Worldwide Inquiries:
Phone: +1.650.506.7000
www.oracle.com

Oracle is committed to developing practices and products that help protect the environment

Hardware and Software, Engineered to Work Together

TABLE OF CONTENTS

List of Tables

List of Figures

1. Introduction

1.1 Overview

This document is the Security Policy for the Acme Packet VME developed by Oracle Communications. Acme Packet VME is also referred to as "the module" or "module". This Security Policy specifies the security rules under which the module shall operate to meet the requirements of FIPS 140-2 Level 1. It also describes how the Acme Packet VME functions to meet the FIPS requirements, and the actions that operators must take to maintain the security of the module.

This Security Policy describes the features and design of the Acme Packet VME module using the terminology contained in the FIPS 140-2 specification. FIPS 140-2, Security Requirements for Cryptographic Module specifies the security requirements that will be satisfied by a cryptographic module utilized within a security system protecting sensitive but unclassified information. The NIST/CCCS Cryptographic Module Validation Program (CMVP) validates cryptographic module to FIPS 140-2. Validated products are accepted by the Federal agencies of both the USA and Canada for the protection of sensitive or designated information.

1.2 Document Organization

The Security Policy document is one document in a FIPS 140-2 Submission Package. The Submission Package contains:

- Oracle Non-Proprietary Security Policy
- Oracle Vendor Evidence document
- Finite State Machine
- Entropy Assessment Document
- Other supporting documentation as additional references

Except for this Non-Proprietary Security Policy, the FIPS 140-2 Validation Documentation is proprietary to Oracle and is releasable only under appropriate non-disclosure agreements. For access to these documents, please contact Oracle.

2. Acme Packet VME

2.1 Functional Overview

The Acme Packet VME is specifically designed to meet the unique price performance and manageability requirements of the small to medium sized enterprise and remote office/ branch office. Ideal for small site border control and Session Initiation Protocol (SIP) trunking service termination applications, the Acme Packet VME deliver Oracle's industry leading ESBC capabilities in binary packaged executable that can be run in a virtual environment.

Acme Packet VME addresses the unique connectivity, security, and control challenges enterprises often encounter when extending real-time voice, video, and UC sessions to smaller sites. The appliance also helps enterprises contain voice transport costs and overcome the unique regulatory compliance challenges associated with IP telephony. An embedded browser based graphical user interface (GUI) simplifies setup and administration.

2.2 FIPS 140-2 Validation Scope

The Acme Packet VME appliances are being validated to overall FIPS 140-2 Level 1 requirements. See Table 1 below.

Security Requirements Section	Level
Cryptographic Module Specification	1
Cryptographic Module Ports and Interfaces	1
Roles and Services and Authentication	2
Finite State Machine Model	1
Physical Security	N/A
Operational Environment	1
Cryptographic Key Management	1
EMI/EMC	1
Self-Tests	1
Design Assurance	3
Mitigation of Other Attacks	N/A

Table 1: FIPS 140-2 Security Requirements

3. Cryptographic Module Specification

3.1 Definition of the Cryptographic Module

The logical cryptographic boundary of the module consists of the Oracle VME OVA image called “nnSCZ900-img-vm_vmware.ova” for version S-Cz9.0.

Figure 1 shows the logical block diagram (red-dotted line) of the module executing in memory and its interactions with the hypervisor through the module’s defined logical cryptographic boundary. The module interacts directly with the hypervisor, which runs directly on the host system.

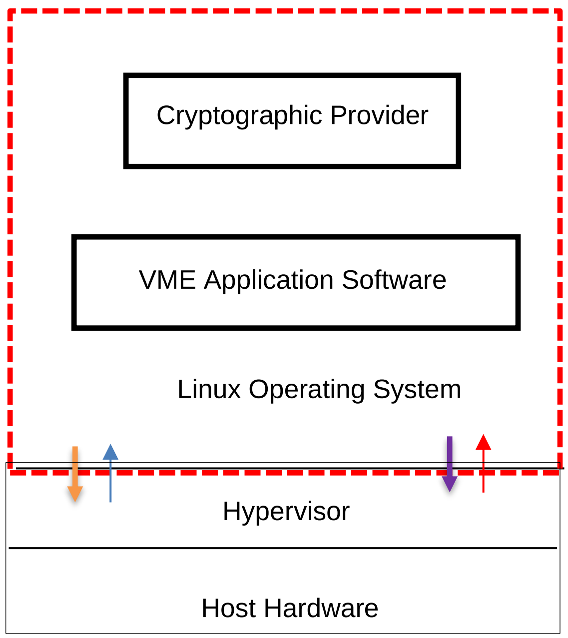
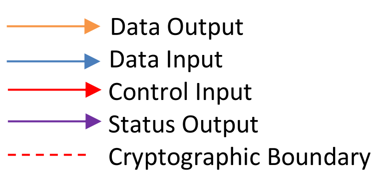


Figure 1: VME Logical Cryptographic Boundary



3.2 Definition of the Physical Cryptographic Boundary

The module consists of binary packaged into an executable that can be run in a virtual environment. The module is classified as a multi-chip standalone cryptographic module. The physical cryptographic boundary is defined as the hard enclosure of the host system on which it runs and no components are excluded from the requirements of FIPS PUB 140-2.

3.3 Approved or Allowed Security Functions

The Acme Packet VME contains the following FIPS Approved Algorithms listed in Table 2 (Oracle Acme Packet Cryptographic Library) and Table 3 (Oracle Acme Packet Mocana Cryptographic Library):

	Approved or Allowed Security Functions	Cert#
Symmetric Algorithms		
AES	CBC, ECB, GCM, GMAC; Encrypt/Decrypt; Key Size = 128, 256 CTR; Encrypt; Key Size = 128,256	A1621
Triple DES[1]	CBC; Encrypt/Decrypt; Key Size = 192	A1621
Secure Hash Standard (SHS)		
SHS	SHA-1[2], SHA-256, SHA-384, SHA-512	A1621
Data Authentication Code		
HMAC	HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	A1621
Asymmetric Algorithms		
RSA	RSA: FIPS186-4: 186-4 KEY(gen): FIPS186-4_Random_e (2048) ALG[ANSIX9.31] SIG(gen) (2048 SHA(256 , 384)), (4096 SHA(256 , 384)) ALG[ANSIX9.31] SIG(Ver) (2048 SHA(1, 256, 384)) RSA: FIPS186-2: ALG[ANSIX9.31] SIG(Ver) (2048 SHA(1, 256, 384)), (4096 SHA (1, 256, 384))	A1621
ECDSA	Firmware: FIPS186-4 KeyGen: (P-256, P-384) SigGen: CURVES (P-256: (SHA-256, 384) P-384: (SHA-256, 384) SigVer: CURVES (P-256: (SHA-256, 384) P-384: (SHA-256, 384))	A1621
Random Number Generation		
DRBG	Firmware: CTR_DRBG: [Prediction Resistance Tested: Not Enabled; BlockCipher_Use_df: (AES-256)]	A1621

[1] Per IG A.13 the same Triple-DES key shall not be used to encrypt more than 2^20 64-bit blocks of data.

[2] SHA-1 is used for SNMP authentication (HMAC-SHA1), SRTP authentication (HMAC-SHA1), SSH Integrity (HMAC-SHA1), TLS integrity (HMAC-SHA1) as defined in Table 2, Table 3 and Table 14, as well as RSA SigVer as defined in Table 2 and Table 3, which are all approved uses for SHA-1.

Key Agreement		
KAS-SSC	KAS-ECC-SSC: Scheme: "Ephemeral Unified" with curve P-256 & P-384 KAS-FFC-SSC: Scheme: "dhEphem" and domain parameter generation method "ffdhe2048"	A1621
KAS	(KAS-SSC Cert. #A1621, CVL Cert. #A1621) IG D.8 Scenario X1 Option 2.	N/A
Key Establishment		
Key Derivation (CVL)	Firmware: SSH KDF, SNMP KDF, SRTP KDF, TLS KDF	(CVL) A1621
Key Transport		
KTS	KTS (AES Cert. #A1621 and HMAC Cert. #A1621; key establishment methodology provides 128 or 256 bits of encryption strength) – AES modes: CBC/CTR/GCM (128-bit and 256-bit). KTS (Triple-DES Cert. #A1621 and HMAC Cert. #A1621; key establishment methodology provides 112 bits of encryption strength)	

Table 2: FIPS Approved and Allowed Security Functions for Oracle Acme Packet Cryptographic Library

	Approved or Allowed Security Functions	Cert #
Symmetric Algorithms		
AES	CBC; Encrypt/Decrypt; Key Size = 128, 192, 256 CTR; Encrypt; Key Size = 128,256	A1620
Triple DES[3]	CBC; Encrypt/Decrypt; Key Size = 192	A1620
Secure Hash Standard (SHS)		
SHS	SHA-1, SHA-256, SHA-384, SHA-512	A1620
Data Authentication Code		
HMAC	HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	A1620
Asymmetric Algorithms		
RSA	RSA: 186-4: 186-4 KEY(gen): FIPS186-4_Random_e (2048) ALG [PKCS1.5]: SIG(Ver) (1024 SHA(1); (2048 SHA (1))	A1620
Key Agreement		
KAS-SSC	KAS-FFC-SSC: Scheme: "dhEphem" and domain parameter generation method "MODP-2048" and "MODP-3072"	A1620
KAS	(KAS-SSC Cert. #A1620, CVL Cert. #A1620) IG D.8 Scenario X1 Option 2.	N/A

[3] Per IG A.13 the same Triple-DES key shall not be used to encrypt more than 2^20 64-bit blocks of data.

Key Establishment		
Key Derivation (CVL)	IKEv1/IKEv2 KDF	(CVL) A1620
Key Transport		
KTS	KTS (AES Cert. #A1620 and HMAC Cert. #A1620; key establishment methodology provides between 128 and 256 bits of encryption strength) – AES modes: CBC (128-bit, 192-bit and 256-bit) and CTR (128-bit and 256-bit). KTS (Triple-DES Cert. #A1620 and HMAC Cert. #A1620; key establishment methodology provides 112 bits of encryption strength)	

Table 3: FIPS Approved and Allowed Security Functions for Oracle Acme Packet Mocana Cryptographic Library

Algorithm	Usage
ENT (NP)	Greater than 256 bits entropy input from the CPU jitter RNG for seeding the SP 800-90A DRBG.

Table 4: Approved SP 800-90B Entropy Source

3.4 Non-Approved But Allowed Security Functions

The following are considered non-Approved but allowed security functions:

Algorithm	Usage
MD5 (TLS 1.2) (no security claimed)	MACing: HMAC, MD5. Hashing: MD5

Table 5: Non-Approved but Allowed Security Functions

3.5 Vendor Affirmed Security Functions

The following services are considered vendor affirmed security functions:

Algorithm	Vendor Affirmed Security Functions
CKG	In accordance with FIPS 140-2 IG D.12, the cryptographic module performs Cryptographic Key Generation (CKG) as per SP800-133rev2 (vendor affirmed). The resulting generated symmetric keys and the seed used in the asymmetric key generation are the unmodified output from an NIST SP 800-90A DRBG.

Table 6: Vendor Affirmed Functions

4. Module Ports and Interfaces

Oracle Virtual Machine edition is a virtualized cryptographic module that meets the overall Level 1 FIPS 140-2 requirements. The module interfaces can be categorized as follows:

- Data Input Interface
- Data Output Interface
- Control Input interface
- Status Output Interface
- Power Interface

The table below provides a mapping of ports for the Oracle VME:

FIPS 140 Interface	Physical Port	VM Port	Logical Interface	Information Input/Output
Data Input	Host System Ethernet (10/100/1000) Ports, Host System USB Ports.	• Virtual Ethernet Ports, • Virtual USB Ports.	API Input Data and Parameters.	Cipher text Plain text
Data Output	Host System Ethernet (10/100/1000) Ports, Host System USB Ports.	• Virtual Ethernet Ports, • Virtual USB Ports.	API Output Data and Parameters.	Cipher text Plain Text
Control Input	Host System Ethernet (10/100/1000) Ports, Host System Serial Ports.	• Virtual Ethernet Ports, • Virtual Serial Ports.	API Command Input Parameters.	• Plaintext control input via console port (configuration commands, operator passwords) • Ciphertext control input via network management (EMS control, CDR accounting, CLI management)
Status Output	Host System Ethernet (10/100/1000) Ports, Host System Serial Ports.	• Virtual Ethernet Ports, • Virtual Serial Ports.	API Status Output Parameters.	Plaintext Status Output via Console Port. Ciphertext Status Output via network management.
Power	Host Power Plug	NA	N/A	N/A

Table 7: Mapping of FIPS 140 Logical interfaces to Logical Ports

5. Physical Security

The module is comprised of software only and thus does not claim any physical security.

6. Roles, Services and Authentication

As required by FIPS 140-2 Level 1, there are three roles (a Crypto Officer Role, User Role, and Unauthenticated Role) in the module that operators may assume. The module supports role-based authentication, and the respective services for each role are described in the following sections. The below table gives a high-level description of all services provided by the module and lists the roles allowed to invoke each service.

Operator Role	Summary of Services
User	• View configuration versions and system performance data • Test pattern rules, local policies, and session translations • Display system alarms.
Crypto-Officer	Allowed access to all system commands and configuration privileges
Unauthenticated	• Request Authentication • Show Status • Initiate self-tests

Table 8: Service Summary

6.1 Operator Services and Descriptions

The below table provides a full description of all services provided by the module and lists the roles allowed to invoke each service.

U	CO	Service Name	Service Description	Keys and CSP(s)	Access Type(s)
	X	Configure	Initializes the module for FIPS mode of operation	HMAC-SHA-256 key	R, W, X
	X	Zeroize CSP's	Clears keys/CSPs from memory and disk	All CSP's	Z
	X	Firmware Update	Updates firmware	Firmware Integrity Key (RSA)	R, X
	X	Bypass	Configure bypass using TCP or UDP and viewing bypass service status	HMAC-SHA-256 Bypass Key	R, W, X

U	CO	Service Name	Service Description	Keys and CSP(s)	Access Type(s)
X	X	Decrypt	Decrypts a block of data Using AES or Triple-DES in FIPS Mode	TLS Session Keys (Triple-DES) TLS Session Keys (AES128) TLS Session Keys (AES256) SSH Session Key (AES128) SSH Session Key (AES256) SRTP Session Key (AES-128) SNMP Privacy Key (AES-128) IKE Session Encryption Key (Triple-DES, AES-128 CBC/CTR, AES-192 CBC, AES-256 CBC/CTR) IPsec Session Encryption Key (Triple-DES, AES-128 CBC/CTR, AES-192 CBC, AES-256 CBC/CTR)	X X X X X X X X X
X	X	Encrypt	Encrypts a block of data Using AES or Triple-DES in FIPS Mode	TLS Session Keys (Triple-DES) TLS Session Keys (AES128) TLS Session Keys (AES256) SSH Session Key (AES128) SSH Session Key (AES256) SRTP Session Key (AES-128) SNMP Privacy Key (AES-128) IKE Session Encryption Key (Triple-DES, AES-128 CBC/CTR, AES-192 CBC, AES-256 CBC/CTR) IPsec Session Encryption Key (Triple-DES, AES-128 CBC/CTR, AES-192 CBC, AES-256 CBC/CTR)	X X X X X X X X X
X	X	Generate Keys	Generates AES or Triple-DES for encrypt/decrypt operations.	TLS Session Keys (Triple-DES) TLS Session Keys (AES128) TLS Session Keys (AES256) SSH Session Key (AES128) SSH Session Key (AES256) SRTP Session Key (AES-128) SNMP Privacy Key (AES-128) IKE Session Encryption Key (Triple-DES, AES-128 CBC/CTR, AES-192 CBC, AES-256 CBC/CTR) IPsec Session Encryption Key (Triple-	R, W R, W R, W R, W R, W R, W R, W R, W

U	CO	Service Name	Service Description	Keys and CSP(s)	Access Type(s)
				DES, AES-128 CBC/CTR, AES-192 CBC, AES-256 CBC/CTR)	R, W
			Generates Diffie-Hellman, EC Diffie-Hellman for key establishment.	Diffie-Hellman Public Key (DH) Diffie-Hellman Private Key (DH) EC Diffie-Hellman Public Key (ECDH) EC Diffie-Hellman Private Key (ECDH) SSH authentication private Key (RSA) SSH authentication public key (RSA) TLS authentication private Key (ECDSA/RSA) TLS authentication public key (ECDSA/RSA) TLS premaster secret, TLS Master secret, SRTP Master key IKE Private Key (RSA) IKE Public Key (RSA) SKEYSEED SKEYID SKEYID_d	R, W R, W R, W R, W R, W R, W R, W R, W R, W R, W R, W R, W R, W R, W R, W R, W
X	X	Verify	Used as part of the TLS, SSH protocol negotiation	SSH authentication private Key (RSA) SSH authentication public key (RSA) TLS authentication private Key (ECDSA/RSA) TLS authentication public key (ECDSA/RSA) Diffie-Hellman Public Key (DH) Diffie-Hellman Private Key (DH) EC Diffie-Hellman Public Key (ECDH) EC Diffie-Hellman Private Key (ECDH)	X X X X X X X X

U	CO	Service Name	Service Description	Keys and CSP(s)	Access Type(s)
X	X	Generate Seed	Generate an entropy_input for CTR DRBG	DRBG Seed DRBG Entropy Input String	R, W, X R, W, X
X	X	Generate Random Number	Generate random number.	DRBG C DRBG V DRBG Key	R, W, X R, W, X R, W, X
X	X	HMAC	Generate HMAC	SNMP Authentication Key SRTP Authentication Key SSH Integrity Keys TLS Integrity Keys IPsec Session Authentication Key IKE Session Authentication Key	X X X X X X
X	X	Generate Certificate	Generate certificate	Web UI Certificate	R, W, X
X	X	Authenticate	Authenticate Users	Operator Password Operator RSA public key	R, W, X R, W, X

R – Read, W – Write, X – Execute, Z - Zeroize

Table 9: Operator Services and Descriptions

6.2 Unauthenticated Services and Descriptions

The below table provides a full description of the unauthenticated services provided by the module:

Service Name	Service Description
On-Demand Self-Test Initialization	This service initiates the FIPS self-test when requested.
Show Status	This service shows the operational status of the module
Authentication	Request authentication to an authorized role.

Table 10: Unauthenticated Services and Descriptions

6.3 Operator Authentication

6.3.1 Password-Based Authentication

In FIPS-approved mode of operation, the module is accessed via Command Line Interface over the Console ports or via SSH or SNMPv3 over the Network Management Ports. The services described are available only to authenticated operators.

Method	Probability of a Single Successful Random Attempt	Probability of a Successful Attempt within a Minute
Password-Based (CO and User Authentication to management interfaces)	Passwords must be a minimum of 8 characters. The password can consist of alphanumeric values, {a-z, A-Z, 0-9, and special characters], yielding 94 choices per character. The probability of a successful random attempt is 1/94^8, which is less than 1/1,000,000.	Passwords must be a minimum of 8 characters. The password can consist of alphanumeric values, {a-z, A-Z, 0-9, and special characters], yielding 94 choices per character Assuming 10 attempts per second via a scripted or automatic attack, the probability of a success with multiple attempts in a one-minute period is 600/94^8, which is less than 1/100,000.
SNMPv3 Passwords	Passwords must be a minimum of 8 characters. The password can consist of alphanumeric values, {a-z, A-Z, 0-9, and special characters], yielding 94 choices per character. The probability of a successful random attempt is 1/94^8, which is less than 1/1,000,000.	Passwords must be a minimum of 8 characters. The password can consist of alphanumeric values, {a-z, A-Z, 0-9, and special characters], yielding 94 choices per character. Assuming 10 attempts per second via a scripted or automatic attack, the probability of a success with multiple attempts in a one-minute period is 600/94^8, which is less than 1/100,000.
Password-Based (SIP Authentication Challenge Response)	Passwords must be a minimum of 12 numeric characters. 0-9, yielding 10 choices per character. The probability of a successful random attempt is 1/10^12, which is less than 1/1,000,000.	Passwords must be a minimum of 12 numeric characters. 0-9, yielding 10 choices per character. Assuming 10 attempts per second via a scripted or automatic attack, the probability of a success with multiple attempts in a one-minute period is 600/10^12, which is less than 1/100,000.

Table 11: Password-Based Authentication

6.3.2 Public Key-Based Authentication

The module also supports public key-based authentication for the Crypto-Officer and User Role with at least 2048-bit RSA keys as implemented by the SSH protocol.

Method	Probability of a Single Successful Random Attempt	Probability of a Successful Attempt within a Minute
Public key-Based	A 2048-bit RSA has at least 112-bits of equivalent strength. The probability of a successful random attempt is 1 /2^112, which is less than 1/1,000,000.	Assuming the module can support 60 authentication attempts in one minute, the probability of a success with multiple consecutive attempts in a one-minute period is 60/2^112, which is less than 1/100,000.

Table 12: Public Key-Based Authentication

7. Key and CSP Management

The following keys, cryptographic key components and other critical security parameters are contained in the module. No parts of the SSH, TLS, IKEv1/IKEv2, SNMP or SRTP protocols, other than the KDF, have been tested by the CAVP and CMVP.

CSP Name	Generation/Input	Establishment/ Export	Storage	Use
Operator Passwords	Generated by the crypto officer as per the module policy	**Agreement:** NA **Entry:** Entry via console or SSH or TLS management session **Output:** Output as part of HA direct physical connection	Virtual Hard Disk	Authentication of the crypto officer and user
Operator RSA public key	Input by the crypto officer and user during the authentication via public keys.	**Agreement:** NA **Entry:** Entry via SSH management session **Output:** N/A	Virtual Hard Disk	Authentication of the crypto officer and user via SSH management session using RSA public keys.
Software Integrity Key (RSA)	Generated externally	**Entry:** RSA (2048 bits) entered as part of Software image **Output:** Output as part of HA direct physical connection	Virtual Hard Disk	Public key used to verify the integrity of software and updates
DRBG Entropy Input String	Generated internally from hardware sources	**Agreement:** NA **Entry:** NA **Output:** None	Volatile RAM	Used in the random bit generation process
DRBG Seed	Generated internally from hardware sources	**Agreement:** NA **Entry:** NA	Volatile RAM	Used in the random bit generation process

CSP Name	Generation/Input	Establishment/ Export	Storage	Use
		Output: None		
DRBG Key	Internal value used as part of SP 800-90A CTR_DRBG	**Agreement:** NA **Entry:** NA **Output:** None	Volatile RAM	Used in the random bit generation process
DRBG V	Internal value used as part of SP 800-90A DRBG	**Agreement:** NA **Entry:** NA **Output:** None	Volatile RAM	Used in the random bit generation process
Diffie-Hellman Public Key (DH) 2048-bit	Internal generation by FIPS-approved CTR_DRBG in firmware	**Agreement:** Diffie-Hellman **Entry:** NA **Output:** None	Volatile RAM	Used to derive the secret session key during DH key agreement protocol
Diffie-Hellman Private Key (DH) 224 bit	Internal generation by FIPS-approved CTR_DRBG	**Agreement:** Diffie-Hellman **Entry:** NA **Output:** None	Volatile RAM	Used to derive the secret session key during DH key agreement protocol
ECDH Public Key (P-256 and P-384)	Internal generation by FIPS-approved CTR_DRBG in firmware	**Agreement:** EC Diffie-Hellman **Entry:** NA **Output:** None	Volatile RAM	Used to derive the secret session key during ECDH key agreement protocol
ECDH Private Key (P-256 and P-384)	Internal generation by FIPS-approved CTR_DRBG	**Agreement**: EC Diffie-Hellman **Entry**: NA **Output**: None	Volatile RAM	Used to derive the secret session key during ECDH key agreement protocol
SNMP Privacy Key	NIST SP 800-135 KDF	**Agreement**: NIST SP 800-135	Volatile RAM	For encryption / decryption of SNMP

CSP Name	Generation/Input	Establishment/ Export	Storage	Use
(AES-128)		KDF **Entry**: NA **Output**: Output as part of HA direct physical connection		session traffic
SNMP Authentication Key (HMAC-SHA1)	Internal generation by FIPS-approved CTR_DRBG in firmware	**Agreement**: NA **Output**: Output as part of HA direct physical connection	Volatile RAM	160-bit HMAC-SHA-1 for message authentication and verification in SNMP
SRTP Master Key (AES-128)	Internal generation by FIPS-approved CTR_DRBG in firmware	**Agreement**: Diffie-Hellman **Entry**: NA **Output**: encrypted or output as part of HA direct physical connection	Volatile RAM	Generation of SRTP session keys
SRTP Session Key (AES-128)	NIST SP 800-135 KDF	**Agreement**: NIST SP 800-135 KDF **Entry**: NA **Output**: Output as part of HA direct physical connection	Volatile RAM	For encryption / decryption of SRTP session traffic
SRTP Authentication Key (HMAC-SHA1)	Derived from the master key	**Agreement**: NA **Output**: Output as part of HA direct physical connection	Volatile RAM	160-bit HMAC-SHA-1 for message authentication and verification in SRTP
SSH Authentication Private Key (RSA)	Internal generation by FIPS-approved CTR_DRBG	**Agreement**: NA **Output**: Output as part of HA direct physical connection	Virtual Hard Disk	RSA private key for SSH authentication
SSH Authentication	Internal generation by FIPS-	**Agreement:** NA	Virtual Hard Disk	RSA public key for SSH authentication.

CSP Name	Generation/Input	Establishment/ Export	Storage	Use
Public Key (RSA)	approved CTR_DRBG	**Output:** Output as part of HA direct physical connection		
SSH Session Keys (AES-128, AES-256)	Derived via SSH KDF. Note: These keys are generated via SSH (IETF RFC 4251). This protocol enforces limits on the number of total possible encryption/decryption operations.	**Agreement**: Diffie-Hellman	Volatile RAM	Encryption and decryption of SSH session
SSH Integrity Keys (HMAC-SHA1)	Derived via SSH KDF.	**Agreement**: NA **Output:** Output as part of HA direct physical connection	Volatile RAM	160-bit HMAC-SHA-1 for message authentication and verification in SSH
TLS Authentication Private Key (ECDSA/RSA)	Internal generation by FIPS-approved CTR_DRBG	**Agreement**: NA **Output:** Output as part of HA direct physical connection	Virtual Hard Disk	ECDSA/RSA private key for TLS authentication
TLS Authentication Public Key (ECDSA/RSA)	Internal generation by FIPS-approved CTR_DRBG	**Agreement**: NA **Output:** Output as part of HA direct physical connection	Volatile RAM	ECDSA/RSA public key for TLS authentication.
TLS Premaster Secret (48 Bytes)	Internal generation by FIPS-approved CTR_DRBG in firmware	**Agreement**: NA **Entry**: Input during TLS negotiation **Output**: NA	Volatile RAM	Establishes TLS master secret
TLS Master Secret (48 Bytes)	Derived from the TLS Pre-Master Secret	**Agreement**: NA	Volatile RAM	Used for computing the Session Key
TLS Session Keys (Triple-DES, AES-128	Derived from the TLS Master Secret	**Agreement**: NA	Volatile RAM	Used for encryption & decryption of TLS session

CSP Name	Generation/Input	Establishment/ Export	Storage	Use
CBC, AES-256)	Note: These keys are generated via TLS (IETF RFC 5246). This protocol enforces limits on the number of total possible encryption/decryption operations.			
TLS Integrity Keys (HMAC-SHA1)	Internal generation by FIPS-approved CTR_DRBG in firmware	**Agreement**: NA **Output**: Output as part of HA direct physical connection	Volatile RAM	160-bit HMAC-SHA-1 for message authentication and verification in TLS
SKEYSEED (20 Bytes)	Derived by using key derivation function defined in SP800-135 KDF (IKEv2).	**Agreement**: NIST SP 800-135 KDF **Entry**: NA **Output**: Output as part of HA direct physical connection to another box	Volatile RAM	160 bit shared secret known only to IKE peers. Used to derive IKE session keys
SKEYID (20 Bytes)	Derived by using key derivation function defined in SP800-135 KDF (IKEv2).	**Agreement**: NIST SP 800-135 KDF **Entry**: NA **Output**: Output as part of HA direct physical connection to another box	Volatile RAM	160 bit secret value used to derive other IKE secrets
SKEYID_d (20 Bytes)	Derived using SKEYID, Diffie Hellman shared secret and other non-secret values through key derivation function defined in	**Agreement**: NIST SP 800-135	Volatile RAM	160 bit secret value used to derive IKE session keys

CSP Name	Generation/Input	Establishment/ Export	Storage	Use
	SP800135 KDF (IKEv1/IKEv2).	KDF **Entry**: NA **Output**: Output as part of HA direct physical connection to another box		
IKE Pre-Shared Key	Preloaded by the Crypto Officer.	**Agreement**: NA **Output**: Output as part of HA direct physical connection to another box	Virtual Hard Disk	Secret used to derive IKE skeyid when using pre-shared secret authentication
IKE Session Encryption Key (Triple-DES, AES-128 CBC/CTR, AES-192 CBC, AES-256 CBC/CTR)	Derived via key derivation function defined in SP800-135 KDF (IKEv1/IKEv2)	**Agreement**: NIST SP 800-135 KDF **Entry**: NA **Output**: Output as part of HA direct physical connection to another box	Volatile RAM	Triple-DES, AES 128 and 256 key used to encrypt data
IKE Session Authentication Key (HMAC-SHA-512)	Derived via key derivation function defined in SP800-135 KDF (IKEv1/IKEv2)	**Agreement**: NIST SP 800-135 KDF **Entry**: NA **Output**: Output as part of HA direct physical connection to another box	Volatile RAM	512 bit key HMAC-SHA-512 used for data authentication
IKE Private Key (RSA 2048 bit)	Internal generation by FIPS-approved CTR_DRBG in firmware	**Agreement**: NA **Output**: Output as part of HA direct physical connection to	Volatile RAM	RSA 2048 bit key used to authenticate the module to a peer during IKE

CSP Name	Generation/Input	Establishment/ Export	Storage	Use
		another box		
IKE Public Key (RSA 2048-bit)	Internal generation by FIPS-approved CTR_DRBG in firmware	**Agreement**: NA **Output:** Output as part of HA direct physical connection to another box	Volatile RAM	RSA 2048 bit public key for TLS authentication.
IPsec Session Encryption Key (Triple-DES, AES-128 CBC/CTR, AES-192 CBC, AES-256 CBC/CTR)	Derived via a key derivation function defined in SP800-135 KDF (IKEv1/IKEv2).	**Agreement**: NIST SP 800-135 KDF **Entry**: NA **Output**: Output as part of HA direct physical connection to another box	Volatile RAM	Triple-DES, AES 128 or 256 bit key used to encrypt data
IPsec Session Authentication Key (HMAC-SHA-512)	Derived via a key derivation function defined in SP800-135 KDF (IKEv1/IKEv2).	**Agreement**: NIST SP 800-135 KDF **Entry**: NA **Output**: Output as part of HA direct physical connection to another box	Volatile RAM	512 bit HMAC-SHA-512 key used for data authentication for IPsec traffic
Web UI Certificate	Internal generation by FIPS approved CTR_DRBG in firmware	**Agreement:** NA **Output:** TLS session with operator	Virtual Hard Disk	Web server certificate
Bypass Key (HMAC-SHA-256)	Internal generation by FIPS-approved CTR_DRBG in firmware	**Agreement**: NA **Output**: NA	Virtual Hard Disk	256-bit HMAC-SHA-256 used to protect bypass table

Table 13: CSP Table

Note: When the module generates symmetric keys or seeds used for generating asymmetric keys, unmodified DRBG output is used as the symmetric key or as the seed for generating the asymmetric keys.

Note: All keys generated by the module use the direct output of a FIPS approved DRBG. This meets the requirements of SP 800-133rev2.

The module employs the Deterministic Random Bit Generator (DRBG) based on [SP800-90A] for the random number generation. The DRBG used for the modules is CTR_DRBG. The module performs the DRBG health tests as defined in section 11.3 of [SP800-90A]. The module uses CPU jitter (inside the physical boundary of the module) as an entropy source for seeding the DRBG. The source is compliant with [SP 800-90B] and marked as ENT(NP) on the certificate. The entropy source is tested with developer defined variants of RCT and APT Health tests as required by section 4 of [SP 800-90B]. The DRBG is seeded with more than 256 bits of entropy strength from the CPU jitter RNG (e.g., 384 bits for the CTR_DRBG using AES-256). Therefore, the module ensures that during initialization (seed) and reseeding, the entropy source provides the required amount of entropy to meet the security strength of the CTR DRBG.

8. Self-Tests

The modules include an array of self-tests that are run during startup and conditionally during operations to prevent any secure data from being released and to ensure all components are functioning correctly. Self-tests may be run on-demand by power cycling the module.

8.1 Power-Up Self-Tests

Acme Packet VME appliance performs the following power-up self-tests when the virtual machine is started. These self-tests require no inputs or actions from the operator:

8.1.1 Software integrity Test

- RSA 2048 Software Integrity Test

8.1.2 Mocana Cryptographic Library Machine Edition (VME) Self-tests

- AES CBC 256-bit (Encrypt/Decrypt) Known Answer Test;
- Triple-DES CBC (Encrypt/Decrypt) Known Answer Test;
- SHA-1 Known Answer Test;
- SHA-256 Known Answer Test;
- SHA-384 Known Answer Test;
- SHA-512 Known Answer Test;
- HMAC-SHA-1 Known Answer Test;
- HMAC-SHA-256 Known Answer Test;
- HMAC-SHA-384 Known Answer Test;
- HMAC-SHA-512 Known Answer Test;
- KAS-FFC-SSC SP800-56arev3 Primitive “Z” Known Answer Test (Modp_2048 & Modp_3072);IKEV1/V2 SP800-135 rev1 KDF Known Answer Test; and
- RSA 2048-bit Signature Verification Test.

8.1.3 Oracle Acme Packet Cryptographic Library Virtual Machine Edition (VME) Self-Tests

- SHA-1 Known Answer Test;
- SHA-256 Known Answer Test;
- SHA-512 Known Answer Test;
- HMAC-SHA-1 Known Answer Test;
- HMAC-SHA-256 Known Answer Test;
- HMAC-SHA-384 Known Answer Test;
- HMAC-SHA-512 Known Answer Test;
- AES ECB 128-bit (Encrypt/Decrypt) Known Answer Test;
- AES GCM 256-bit (Encrypt/Decrypt) Known Answer Test;
- Triple-DES CBC (Encrypt/Decrypt) Known Answer Test;
- SP 800-90A CTR DRBG Known Answer Test;
- KAS-FFC-SSC SP800-56arev3 Primitive “Z” Known Answer Test (Modp_2048);
- KAS-ECC-SSC SP800-56arev3 Primitive “Z” Known Answer Test (P256 & P384);
- SP800-135 KDF Know Answer Tests: SSH KDF, TLS KDF, SNMP KDF and SRTP KDF;
- RSA 2048-bit sign/verify Known Answer Test; and
- ECDSA P-256 sign/verify PCT.

8.1.4 SP 800-90B Health Tests

- APT and RCT Start-up tests. (The start-up tests are the continuous tests run on the first 1024 samples)

When the module is in a power-up self-test state or error state, the data output interface is inhibited and remains inhibited until the module can transition into an operational state. While the CO may attempt to restart the module to clear an error, the module will require re-installation in the event of a hard error such as a failed self-test.

8.2 Critical Functions Self-Tests

Acme Packet VME performs the following critical self-tests. These critical function tests are performed for each SP 800-90A DRBG implemented within the module.

- SP 800-90A Instantiation Test
- SP 800-90A Generate Test
- SP 800-90A Reseed Test
- SP 800-90A Uninstantiate Test

8.3 Conditional Self-Tests

The module performs the following conditional self-tests when called by the module:

- Pair Wise consistency tests to verify that the asymmetric keys generated for RSA, and ECDSA work correctly by performing a sign and verify operation;
- Continuous Random Number Generator test to verify that the output of approved-DRBG is not the same as the previously generated value;
- Developer defined variants of Repetition Count Test (RCT) and Adaptive Proportion Test (APT) are run on the output of noise source that are SP800-90B compliant to verify the output of the noise source;
- Bypass conditional test using HMAC-SHA-256 to ensure the mechanism governing media traffic is functioning correctly, and;
- Software Load test using a 2048-bit/SHA-256 RSA-Based integrity test to verify software to be updated.

9. Crypto-Officer and User Guidance

This section describes the configuration, maintenance, and administration of the cryptographic module. If the steps outlined in Section 9.1 below are not followed, the module will be operating in a non-compliant state that is out of scope of the validation.

9.1 Secure Setup for FIPS Mode of Operation

FIPS Mode is enabled by a license installed by Oracle, which will open the FIPS self-test feature, and implementing the following steps:

1. Open CLI: type "setup entitlements"
2. Select "5 Data Integrity (FIPS 140-2)" option and type "enabled"
3. Type "s" to save the above modified entitlements.
4. Then reboot the module for FIPS mode to take into effect.
5. Then from a CLI the operator must enable FIPS by selecting the FIPS 140-2 option and typing "enabled" and then reboot the device.

Once the secure setup and the secure initialization and configuration is complete the module is in FIPS mode. The steps outlined in 9.1.1 can be performed to ensure that the FIPS Approved mode was correctly configured.

9.1.1 Secure Initialization and Configuration Verification Steps

The crypto-officer can verify FIPS settings by following these steps:

- Verify that the firmware version of the module is Version S-Cz9.0 ("show version" section in Session Border Controller ACLI Configuration Guide (SBC Guide).
- A new account for the Crypto-Officer and User shall be created as part of Setup and Initialization process. Upon creation of the new CO and User accounts the "default" accounts shipped with the module shall be disabled ("local-accounts" section in SBC Guide).
- Ensure all management traffic is encapsulated within a trusted session by encapsulating in a TLS, SSH, or SRTP tunnel as appropriate ("TLS-profile", "SSH-config" and "Sdes-profile" sections in SBC Guide).
- HTTPS shall be enabled and configure the web server certificate prior to connecting to the WebUI over TLS ("http-config" section in SBC Guide).
- Ensure that SNMP V3 is configured with AES-128/HMAC only ("SNMP-Group-Entry" section in SBC Guide).
- Ensure IKEv1 and IKEv2 is using AES CBC or CTR mode for encryption and HMAC-SHA-512 for authentication ("IKE-sainfo" section in SBC Guide).
- Ensure SSH is configured to use AES CTR mode for encryption ("Configure SSH Ciphers" section in SBC Guide).
- Ensure SSH and IKEv1/IKEv2 only use Diffie-Hellman group 14 in FIPS approved mode ("IKE-config" & "SSH-config" section in SBC Guide).
- Ensure RSA keys are at least 2048-bit keys for TLS, IKEv1/IKEv2. No 512-bit or 1024-bit keys can be used in FIPS mode of operation ("Certificate-record" section in SBC Guide - 2048 is the default RSA modulus).
- All operator passwords must be a minimum of 8 characters in length ("password-policy" section in SBC Guide).
- Ensure use of FIPS-approved algorithms for TLS ("TLS-profile" in SBC Guide):
  - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

  - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
  - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
  - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
  - TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
  - TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
  - TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
  - TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
  - TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
  - TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
- Be aware that when configuring High Availability (HA), only a local HA configuration to a directly connected box via a physical cable over the management port is allowed in FIPS Approved Mode. Remote HA is not allowed in FIPS Approved mode.
- Be aware that HA configuration data that contains keys and CSP's must never be transported over an untrusted network. Ensure that the HA ports used for the transport of HA data (including keys and CSP's) are bound to a private IP address range during setup.
- Be aware that only the HA state transactions between the two devices over the direct physical connection are permitted over those dedicated ports.
- RADIUS and TACACS+ shall not be used in FIPS approved mode.
- Any firmware loaded into this module that is not shown on the module certificate, is out of the scope of this validation and requires a separate FIPS 140-2 validation.
- 3-key Triple-DES has been implemented in the module and is FIPS approved until December 31, 2023. Should the CMVP disallow the usage of Triple-DES post-December 31, 2023, then users must not configure Triple-DES.

For more details please refer to the Session Border Controller ACLI Configuration Guide (SBC Guide)

9.1.2 AES-GCM IV Construction/Usage

The AES-GCM IV is used in the following protocols:

- TLS: The TLS AES-GCM IV is generated in compliance with TLSv1.2 GCM cipher suites as specified in RFC 5288 and section 3.3.1 of NIST SP 800-52rev1. Per RFC 5246, when the nonce_explicit part of the IV exhausts the maximum number of possible values for a given session key, the module will trigger a handshake to establish a new encryption key.

In case the module's power is lost and then restored, the key used for the AES GCM encryption or decryption shall be redistributed.

9.2 Disabling FIPS Mode of Operation

FIPS Approved Mode of operation is disabled by uninstalling the Session Border Controller software (which contains the Acme Packet VME module) from the host machine. The module is also considered in non-FIPS mode if the steps in 9.1, 9.1.1 are not performed.

10. Mitigation of Other Attacks

The module does not mitigate attacks beyond those identified in FIPS 140-2.

11. Operational Environment

11.1 Tested Environments

The module is installed using a common base image distributed in a compatible hypervisor format (i.e ova, ovm, qcow2). The software image that is used to deploy the VME is common across all models. The tested configuration includes:

Operating Environment	Processor	Hardware
Oracle Linux 7 running on VMware ESXi version 6.7	Intel Xeon Platinum 8260 Processor	Oracle Server X8-2

Table 14: Operating environment

This is considered a modifiable OE as defined by FIPS 140-2. The tested operating environments isolate virtual systems into separate isolated process spaces. Each process space is logically separated from all other processes by the operating environments software and hardware. The module functions entirely within the process space of the isolated system as managed by the single operational environment. This implicitly meets the FIPS 140-2 requirement that only one entity at a time can use the cryptographic module.

11.2 Vendor Affirmed Environment

The following platforms have not been tested as part of the FIPS 140-2 level 1 certification however Oracle "vendor affirms" that these platforms are equivalent to the tested and validated platform. Additionally, Oracle affirms that the module will function the same way and provide the same security services on the system listed below.

Operating Environment	Processor	Hardware
Oracle Linux 7 running on VMware ESXi version 6.7	Intel Xeon Platinum Processor	Oracle Server X7-2
Oracle Linux 7 running on VMware ESXi version 6.7	Intel Xeon Processor E5-2600 V3	Oracle Server X5-2

Table 15: Vendor Affirmed Operating Environment

CMVP makes no statement as to the correct operation of the module or the security strengths of the generated keys when so ported if the specific operational environment is not listed on the validation certificate.

Acronyms, Terms and Abbreviations

Term	Definition
ACLI	Acme Command Line Interface
AES	Advanced Encryption Standard
CDR	Call Data Record
CMVP	Cryptographic Module Validation Program
CSEC	Communications Security Establishment Canada
CSP	Critical Security Parameter
DHE	Diffie-Hellman Ephemeral
DRBG	Deterministic Random Bit Generator
ECDSA	Elliptic Curve Digital Signature Algorithm
ESBC	Enterprise Session Border Controller
EDC	Error Detection Code
EMS	Enterprise Management Server
HMAC	(Keyed) Hash Message Authentication Code
IKE	Internet Key Exchange
KAT	Known Answer Test
KDF	Key Derivation Function
LED	Light Emitting Diode
MGT	Management
NIST	National Institute of Standards and Technology
POST	Power On Self Test
PUB	Publication
RAM	Random Access Memory
ROM	Read Only Memory
SHA	Secure Hash Algorithm
SNMP	Simple Network Management Protocol
SRTP	Secure Real Time Protocol
TDM	Time Division Multiplexing
TLS	Transport Layer Security
VME	Virtual Machine Edition

Table 16: Acronyms

ORACLE

References

The FIPS 140-2 standard, and information on the CMVP, can be found at http://csrc.nist.gov/groups/STM/cmvp/index.html.

More information describing the module can be found on the Oracle web site at https://docs.oracle.com/en/industries/communications/session-border-controller/index.html.

This Security Policy contains non-proprietary information. All other documentation submitted for FIPS 140-2 conformance testing and validation is "Oracle - Proprietary" and is releasable only under appropriate non-disclosure agreements.

Document	Author	Title
FIPS PUB 140-2	NIST	FIPS PUB 140-2: Security Requirements for Cryptographic Modules
FIPS IG	NIST	Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program
FIPS PUB 140-2 Annex A	NIST	FIPS 140-2 Annex A: Approved Security Functions
FIPS PUB 140-2 Annex B	NIST	FIPS 140-2 Annex B: Approved Protection Profiles
FIPS PUB 140-2 Annex C	NIST	FIPS 140-2 Annex C: Approved Random Number Generators
FIPS PUB 140-2 Annex D	NIST	FIPS 140-2 Annex D: Approved Key Establishment Techniques
DTR for FIPS PUB 140-2	NIST	Derived Test Requirements (DTR) for FIPS PUB 140-2, Security Requirements for Cryptographic Modules
NIST SP 800-67	NIST	Recommendation for the Triple Data Encryption Algorithm TDEA Block Cypher
FIPS PUB 197	NIST	Advanced Encryption Standard
FIPS PUB 198-1	NIST	The Keyed Hash Message Authentication Code (HMAC)
FIPS PUB 186-4	NIST	Digital Signature Standard (DSS)
FIPS PUB 180-4	NIST	Secure Hash Standard (SHS)
NIST SP 800-131A	NIST	Recommendation for the Transitioning of Cryptographic Algorithms and Key Sizes
PKCS#1	RSA Laboratories	PKCS#1 v2.1: RSA Cryptographic Standard

Table 17: References