# Secure Computing Corporation

# SafeWord SecureWire 500 and SafeWord SecureWire 2500 Identity and Access Management Appliances

(Firmware version: R2.6.0, Hardware Version: 500 Rev 100-000001, 2500 Rev 100-000002)



# FIPS 140-2
# Non-Proprietary Security Policy

**Level 2 Validation**

**Document Version 1.0**

Prepared for:

**Secure Computing Corporation**
1855 Gateway Boulevard, Suite 200
Concord, CA 94520
Phone: (408) 979-6100
Fax: (408) 979-6501
http://securecomputing.com/

Prepared by:

**Corsec Security, Inc.**
10340 Democracy Lane, Suite 201
Fairfax, VA 22030
Phone: (703) 267-6050
Fax: (703) 267-6810
http://www.corsec.com

# Revision History

| Version | Modification Date | Modified By | Description of Changes |
|---|---|---|---|
| 0.1 | 2006-05-02 | Rumman Mahmud | Initial draft. |
| 0.2 | 2006-05-04 | Rumman Mahmud | Draft |
| 0.2 | 2006-06-06 | Rumman Mahmud | Revised draft |
| 1.0 | 2006-06-29 | Rumman Mahmud | Final draft |

# Table of Contents



# Table of Figures

# Table of Tables

# 0 Introduction

## 0.1 Purpose

This is a non-proprietary Cryptographic Module Security Policy for the SafeWord SecureWire 500 and SafeWord SecureWire 2500 Identity and Access Management Appliances from Secure Computing Corporation. This Security Policy describes how the SafeWord SecureWire 500 and SafeWord SecureWire 2500 Identity and Access Management Appliances meets the security requirements of FIPS 140-2 and how to run the module in a secure FIPS 140-2 mode. This policy was prepared as part of the Level 2 FIPS 140-2 validation of the module.

FIPS 140-2 (Federal Information Processing Standards Publication 140-2 – *Security Requirements for Cryptographic Modules*) details the U.S. Government requirements for cryptographic modules. More information about the FIPS 140-2 standard and validation program is available on the National Institute of Standards and Technology (NIST) Cryptographic Module Validation Program (CMVP) website at: http://csrc.nist.gov/cryptval/

The SafeWord SecureWire 500 and SafeWord SecureWire 2500 Identity and Access Management Appliances are referred to in this document as the 500 and 2500, the appliances, or the modules.

## 0.2 References

This document deals only with operations and capabilities of the module in the technical terms of a FIPS 140-2 cryptographic module security policy. More information is available on the module from the following sources:

- The Secure Computing website (http://securecomputing.com) contains information on the full line of products from Secure Computing.
- The CMVP website (http://csrc.nist.gov/cryptval/) contains contact information for answers to technical or sales-related questions for the module.

## 0.3 Document Organization

The Security Policy document is one document in a FIPS 140-2 Submission Package. In addition to this document, the Submission Package contains:

- Vendor Evidence document
- Finite State Machine
- Other supporting documentation as additional references

This Security Policy and the other validation submission documentation were produced by Corsec Security, Inc. under contract to Secure Computing. With the exception of this Non-Proprietary Security Policy, the FIPS 140-2 Validation Documentation is proprietary to Secure Computing and is releasable only under appropriate non-disclosure agreements. For access to these documents, please contact Secure Computing.

# 1 SafeWord SecureWire 500 and SafeWord SecureWire 2500 Identity and Access Management Appliances

## 1.1 Overview

Secure Computing® has been securing the connections between people and information for years, specializing in creating a trusted environment both inside and outside organizations. SafeWord® SecureWire™ is a powerful identity and access management (IAM) appliance that secures access, enforces policy, and provides complete and customizable reporting for entire network. SecureWire provides lightning fast, ultra secure access to every application and data resource in network – with identity, security, and simplicity in mind. As a vital component of complete identity and access management strategy, SecureWire revolutionizes the way to provide access to employees, business partners, and extranet users.

With SecureWire, all methods are hosted on a single appliance, greatly simplifying rollout and enforcement of security policy changes. SecureWire allows to segment network into logical security zones, based upon the sensitivity of the resource. Only properly identified users with secure devices and the proper level of authentication can access these zones, which require no reconfiguration of network infrastructure. A trusted machine inside the building may have full access rights to all applications, but a remote device such as a home PC may be restricted to only Webmail or view-only rights.

With access setup, policy enforcement, user management, and configuration compliance centralized on SecureWire, reporting and auditing can be simplified dramatically. Every access, authentication, and authorization request can be reported—on the LAN, over the VPN, over the Web, across every access point in network. SecureWire consolidates reporting on a single tool to create a single report. Not only does this cut down on effort with auditing, reporting, and compliance, it also reduces errors integrating disparate reports.

The SafeWord SecureWire 500 and SafeWord SecureWire 2500 Identity and Access Management Appliances provide endpoint security, strong authentication and single point for policy enforcement and reporting. These appliances are validated at the following FIPS 140-2 Section levels:

**Table 1 - Security Level Per FIPS 140-2 Section**

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

## 1.2 Module Interfaces

The SafeWord SecureWire 500 and SafeWord SecureWire 2500 Identity and Access Management Appliances are multi-chip standalone modules that meet overall level 2 FIPS 140-2 requirements. The 500 is a 1U rack-mountable server, and the 2500 is a 2U rack-mountable service. Both devices are completely enclosed in a hard, opaque metal case with tamper-evident labels, and this case is defined as the cryptographic boundary of the modules.

SecureWire 2500 and SecureWire 500 provide specific physical ports that cross the cryptographic boundary of the devices, and these ports provide the only access to the module's services.

The SwcureWire 2500 provides the following physical ports:

- Management port (10/100 Ethernet port - RJ45 connector) dedicated to management
    - The link LED (lower left corner of the (RJ45) Ethernet connectors) for each of the ports is only lit when the Ethernet port has an active link, and the act LED (lower left corner of the (RJ45) Ethernet connectors) blinks when traffic is flowing over the port.
- 4 LAN ports (Gigabit Ethernet ports - RJ45 connectors) for data path (redundant pairs) and optionally management
    - The link LED (lower left corner of the (RJ45) Ethernet connectors) for each of the ports is only lit when the Ethernet port has an active link, and the act LED (upper left corner of the (RJ45) Ethernet connectors) blinks when traffic is flowing over the port.
- 1 Serial port (RJ45 connector) for management
- 1 SDM (LCD display, status LEDs, buttons) panel for limited management
- 1 power button
- 2 power connectors
- 2 USB ports for high availability (HA) – management

The following is a list of the physical ports for the 500:

- 2 LAN ports (Gigabit Ethernet ports – RJ45 connectors) for data path and management
- The link LED (lower left corner of the (RJ45) Ethernet connectors) for each of the ports is only lit when the Ethernet port has an active link, and the act LED (upper left corner of the (RJ45) Ethernet connectors) blinks when traffic is flowing over the port.
- 1 Serial port for management
- 1 SDM (LCD display, status LEDs, buttons) panel for limited management
- 1 power button
- 1 power switch
- 1 power connector
- 2 USB ports for HA – management

The physical ports of the 500 and 2500 are depicted in the following figures.

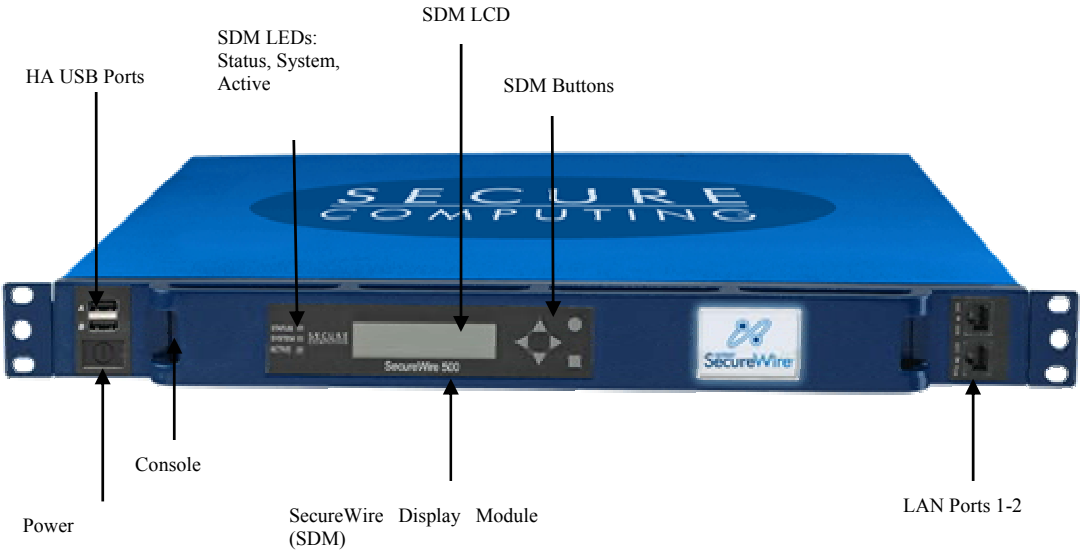


**Figure 1 – SecureWire 500 Front Panel**


**Figure 2 – SecureWire 500 Rear Physical Ports**

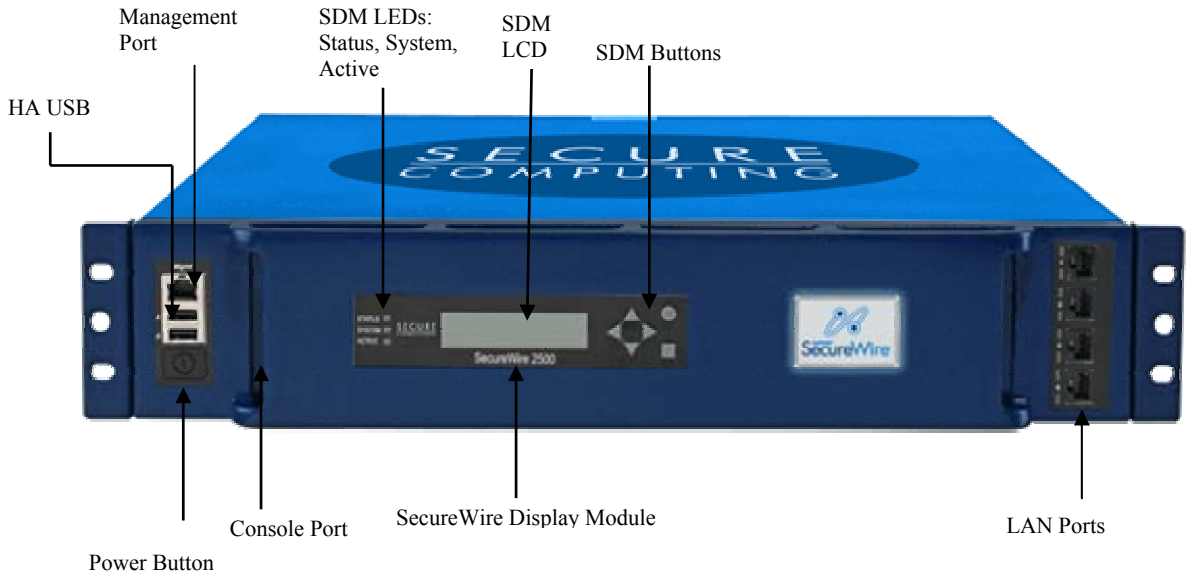


**Figure 3 - SecureWire 2500 Front Panel**

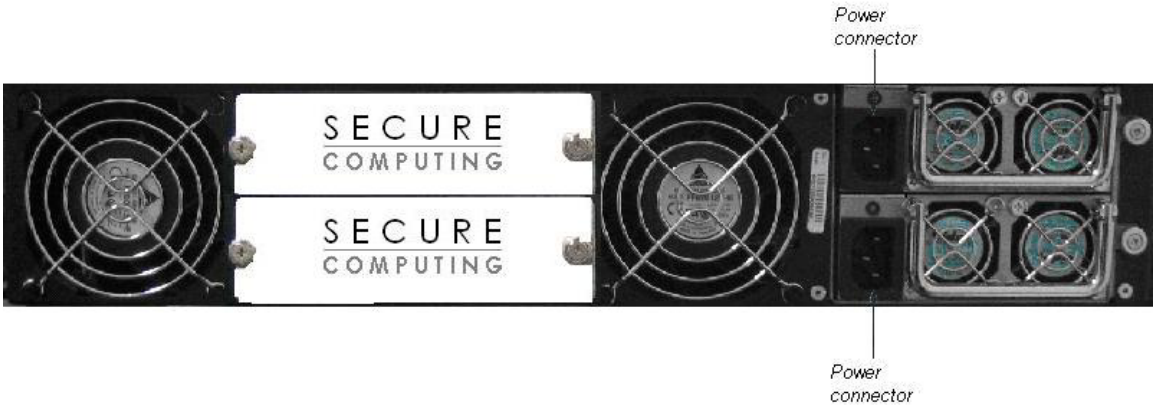


**Figure 4 - SecureWire 2500 Rear Physical Ports**

The LEDs of the SDM indicate the following status.

**Table 2 - SDM Status Indicator LEDs**

| LED Name | Color | Description |
|---|---|---|
| SYSTEM | Green | All diagnostics have passed and the system is operational. |
| | Orange (Amber) | Flashing: The system is booting and/or running diagnostics (normal initialization sequence). |
| | Red | The system is not operational because a fault has occurred during the initialization sequence. |
| | Off | No Power |
| STATUS | Green | No alarms. |
| | Orange (Amber) | Minor Alarm(s) are present on the system. Minor alarms need to be defined, but may include various chassis environmental, power supplies, and fan monitors. |

| | | |
|---|---|---|
| | Red | Major or critical alarm(s) are present on the system. |
| | Off | No Power |
| ACTIVE | Green | The system is in Active mode. |
| | Orange (Amber) | The system is in Standby mode. |
| | Red | Active/Standby Failure, or Not Ready |
| | Off | Inactive |

The physical ports of the 500 and 2500 map to the logical interfaces of FIPS 140-2, as described in the following table.

**Table 3 - Physical Ports and Logical Interfaces**

| 2500 PHYSICAL PORT | 500 PHYSICAL PORT | FIPS 140-2 LOGICAL INTERFACE |
|---|---|---|
| Power connectors | Power connector | Power interface |
| 10/100 BaseT Ethernet port (dedicated management) | | Data input, data output, control input, status output |
| 10/100/1000 BaseT Ethernet ports (LAN, WAN, management) | 10/100/1000 BaseT Ethernet ports (LAN, WAN, management) | Data input, data output, Control input, status output |
| RJ45 serial port (CLI) | RJ45 serial port (CLI) | Control input, status output |
| USB ports | USB ports | Data input, data output |
| LEDs | LEDs | Status output |
| LCD | LCD | Status Output |
| LCD Buttons | LCD Buttons | Control input |
| Power Button | Power Button | Control input |
| | Hard power switch | Control input |

## 1.3 Roles and Services

SecureWire 500 and 2500 support three roles: a Crypto-Officer, a User, and a Network User. The module supports identity-based authentication.

### 1.3.1 Crypto Officer Role

The Crypto-Officer can manage the SecureWire modules over a TLSv1 session using SWCC API calls through a software Graphical User Interface (GUI) application provided by Secure Computing called the SecureWire Control Center (SWCC). Through this interface, the Crypto-Officer is able to configure Users of the device, load/generate key pairs, configure IPSec settings, and perform virtually all of the management of the module.

Additionally, Crypto-Officers can manage the box using a Command Line Interface (CLI) over the locally connected serial port or remotely via SSHv2. The CLI contains a subset of the remote management functionality provided through the SWCC API and some additional commands, most notably software upgrading.

Crypto-Officers can be defined with different subsets of functionality (user, system, superuser, monitor, security, user-defined), and these subsets are/can be configured with varying degrees of hide/read/write permissions for administrative functionality. Only the “admin” superuser is able to access the CLI.

Crypto-Officer service descriptions are provided in the table below.

**Table 4 – Mapping of Crypto Officer Role's Services to Inputs, Outputs, CSPs, and Type of Access**

| Service | Description | Input | Output | CSP and Type of Access |
|---|---|---|---|---|
| Login | Authenticate the Crypto-Officer role, which can be on top of authentication mechanism employed by cryptographic protocols. | Login information | Result of login attempt | Crypto-Officer password - Read |
| Logout | Log out the Crypto-Officer. | Logout call | Call response | Logout |
| administer | Access administer level sub-commands. | Command | Status | |
| clear-known-hosts | Clear the database of known host keys. | Administer sub-command and parameters | Status | |
| dumpstate | Dump state information to log. | Administer sub-command and parameters | Status and state information | |
| file | File operation utilities. | Administer sub-command and parameters | Status information and file data | |
| ha | High Availability information and administration. | Administer sub-command, HA sub-commnds, if necessary, and parameters, including HA shared secret, if configuring | Status information | HA shared secret – Write |
| log | Log utility. | Administer sub-command and parameters | Status information | |
| reboot | Reboot the system. | Administer sub-command | Status | |
| release | Firmware upgrade. | Administer sub-command and parameters, including firmware release if uploading to box | Status information | Firmware upgrade public key – Read/Write |
| secpassword | Set boot password. | Administer sub-command and parameters, including a password (not a critical security parameter), if configuring | Status information | |

| Service | Description | Input | Output | CSP and Type of Access |
|---|---|---|---|---|
| show | Display administrator settings. | Administer sub-command | Status information | |
| shutdown | Power down of system. | Administer sub-command | Status | |
| sysfailaction | Set system failure action. | Administer sub-command and parameters | Status | |
| timeout | Set the CLI idle timeout period. | Administer sub-command and parameters | Status | |
| userid | User configuration. | Administer sub-command, userid sub-command, if necessary, and parameters | Status information | |
| arp | Address resolution display and control | Sub-command and applicable parameters | Status information | |
| configure | General system configuration commands. | Command and parameters | Status information | |
| diagnostics | General system diagnostics. | Command and parameters | Status information | |
| exit | Exit from CLI interface. | Command | Status | |
| help | Display context sensitive help text. | Command and parameters | Status information | |
| ping | Send ICMP ECHO_REQUEST packets to network hosts. | Command and parameters | Status information | |
| quit | Exit from CLI interface. | Command | Status | |
| saveconfig | Save the current configuration database. | Command | Status | |
| show | Show system information. | Command and parameters | Status information | |
| top | Move to top-most level of command hierarchy. | Command | Status | |

| Service | Description | Input | Output | CSP and Type of Access |
|---|---|---|---|---|
| traceroute | Print the route packets take to a network host. | Command and parameters | Status information | |
| up | Move one level up the command hierarchy. | Command | Status | |
| Open CLI | Establish an SSH session and authenticate an operator with digital certificates, if configured. | SSH handshake parameters, SSH inputs | SSH outputs | SSH session keys - Read/Write<br><br>DSA/RSA private keys - Read<br><br>DSA/RSA public keys - Read/Write |
| Policy | Configure access control policies for Users, Web/File Resources, Applications, Machines, and Networks. | SWCC API calls and configuration information, including User passwords, if configuring | Status information, including configured policies | User passwords – Read/Write |
| App Protection | Configure IDS rules for monitoring traffic | SWCC API calls and configuration information | Status information, including configured policies | |
| Authentication | Configuration authentication policies, including password length and complexity restrictions | SWCC API calls and configuration information | Status information, including configured policies | |
| Configure SecureWire Connect | Configure SecureWire Connect functionality | SWCC API calls and configuration information | Status information, including configured policies | |
| IPSec | Configure the IPSec services, including setting up whether to use IKE, what algorithms to support, what keys and certificates to use, etc. | SWCC API calls and configuration information, including pre-shared keys and manually configured IPSec session keys, depending on configuration | Status information, including configured policies | Pre-shared keys – Write<br><br>IPSec session keys - Write |

| Service | Description | Input | Output | CSP and Type of Access |
|---|---|---|---|---|
| Configure system settings | Configure the system settings, including generation of key pairs, configuring certificates, setting up SSL servers | SWCC API calls and configuration information, including public keys, depending on configuration | Status information, including configured policies | RSA/DSA public keys – Read/Write<br><br>RSA/DSA private keys – Read/Write |
| Administration | Configure administrators, including Crypto-Officer passwords, and monitor the devices | SWCC API calls and configuration information, including Crypto-Officer passwords, if configuring | Status information, including configured policies | Crypto-Officer passwords – Read/Write |
| Zeroization | Zeroize the module's CSPs | SWCC API calls | Status information | ALL CSPs - Write |
| TLS | Establish a TLS session. | TLS handshake parameters, TLS inputs | TLS outputs | TLS session keys - Read/Write<br><br>RSA private keys - Read<br><br>RSA public keys - Read/Write |

### 1.3.2 User Role

Users authenticate to the SecureWire devices and are granted access to particular resources (networks, servers, and files) based on the access control permission configured for that operator or resource. These access control permissions are configured by administrators, and this configuration is an extremely robust, policy based architecture.

The Users are able to access the module over a TLS session or through an IPSec tunnel. On top of these protocols, users authenticate using user IDs (UIDs) and passwords, and the authentication may be performed internally using a local database or via a 3rd party authentication mechanism, such as Radius.

User service descriptions and inputs/outputs are listed in the following table:

**Table 5 – Mapping of User Role's Services to Inputs, Outputs, CSPs, and Type of Access**

| Service | Description | Input | Output | CSP and Type of Access |
|---|---|---|---|---|
| Login | Authenticate the User role, which can be on top of authentication mechanism employed by cryptographic protocols. | Login information | Result of login attempt | User password - Read |
| Logout | Log out the User. | Logout call | Call response | |

| Service | Description | Input | Output | CSP and Type of Access |
|---|---|---|---|---|
| Password change | Change the User's password | Password change information | Status | User password - Write |
| Resource, Application, and Network access | Users, Web/File Resources, Applications, Machines, and Networks. | Data inputs for particular resource, application, or network | Data outputs for particular resource, application, or network | |
| TLS | Establish a TLS session. | TLS handshake parameters, TLS inputs | TLS outputs | TLS session keys - Read/Write<br><br>DSA/RSA private keys - Read<br><br>DSA/RSA public keys - Read/Write |
| IPSec | Establish an IPSec session and authenticate an operator with digital certificates or pre-shared, if configured. | IPSec handshake parameters, IPSec inputs | IPSec outputs | IPSec session keys - Read/Write<br><br>Pre-shared keys - Read<br><br>DSA/RSA private keys - Read<br><br>DSA/RSA public keys - Read/Write |

### 1.3.3 Network User Role

Network users (networks, machines) authenticate to the SecureWire devices and are granted access to particular resources (networks, servers, files) based on the access control permission configured for that network. These access control permissions are configured by administrators, and this configuration is an extremely robust, policy based architecture.

The Network Users are able to access the module through an IPSec tunnel, which authenticates Network Users via pre-shared keys or digital certificates.

Network User service descriptions and inputs/outputs are listed in the following table:

**Table 6 - Network User Services, Descriptions, Inputs and Outputs**

| Service | Description | Input | Output | CSP and Type of Access |
|---|---|---|---|---|
| Resource, Application, and Network access | Users, Web/File Resources, Applications, Machines, and Networks. | Data inputs for particular resource, application, or network | Data outputs for particular resource, application, or network | |

| Service | Description | Input | Output | CSP and Type of Access |
|---|---|---|---|---|
| IPSec | Establish an IPSec session and authenticate an operator with digital certificates or pre-shared, if configured. | IPSec handshake parameters, IPSec inputs | IPSec outputs | IPSec session keys - Read/Write<br><br>Pre-shared keys - Read<br><br>DSA/RSA private keys - Read<br><br>DSA/RSA public keys - Read/Write |

### 1.3.4 Authentication Mechanisms

The Crypto-Officers are able to access the module over a TLS or SSH session, or through a directly connected console port. On top of these protocols, Crypto-Officers authenticate using user IDs (UIDs) and passwords. The Users are able to access the module over a TLS session or through an IPSec tunnel. On top of these mechanisms, Users authenticate using user IDs (UIDs) and passwords, and the authentication may be performed internally using a local database or via a 3rd party authentication mechanism, such as Radius 3rd party. Network Users authenticate during IPSec via pre-shared keys or digital certificates.

**Table 7 - Authentication Mechanisms**

| Authentication Type | Strength |
|---|---|
| Passwords | Considering a case sensitive alphanumeric password with repetition, the total possible combinations for the password are 62^6. The probability for a random attempt to succeed is 1:62^6 or 1:56800235584, and, since this authentication attempt is additionally piped over an encrypted session, it is not possible to perform enough authentication attempts to reduce the 1:62^6 chance per attempt to 1:100,000 over a minute. |
| Pre-shared Keys | Considering a case sensitive alphanumeric pre-shared key with repetition, the total possible combinations for the pre-shared key are 62^16. The probability for a random attempt to succeed is 1:62^16 or 1: 4767240170682353345026333081 6, and it is not possible for an operator to perform enough authentication attempts to reduce the 1: 62^16 chance per attempt to 1:100,000 over a minute. |
| Public key certificates | Using conservative estimates equating a 1024 bit DSA/RSA key to an 80 bit symmetric key, the probability for a random attempt to succeed is 1:2^80 or 1:1208925819614629174706176, and it is not possible for an operator to perform enough authentication attempts to reduce the 1: 2^80 chance per attempt to 1:100,000 over a minute |

### 1.3.5 Unauthenticated Services

The module has a limited set of unauthenticated services, which are only available through local access to the module using the SDM. (Although a password can be configured for authentication through the SDM during boot, this is not required nor deemed security-relevant as the services provided by the SDM are meant to be unauthenticated and FIPS 140-2 does not require a password to be entered for booting a module.)

**Table 8 - Unauthenticated Services, Descriptions, Inputs and Outputs**

| Service | Description | Input | Output | CSP and Type of Access |
|---|---|---|---|---|
| SDM status | Status information displayed on the SDM | | Status | |
| Factory defaults (zeroization) | Zeroize all CSPs and return to factory default state | SDM button input | Command status | All CSPs - Write |
| Management network interface configuration | Set the network settings for the management port | SDM button input | Command status | |
| Savecore | Saves the system core | SDM button input | Command status | |
| Restart | Reboots the system | SDM button input | Command confirmation and status | |
| Viewing alarms | View system alarms | SDM button input | Command status | |
| Shutdown | Shuts down the system | SDM button input | Command confirmation and status | |

## 1.4 Physical Security

SafeWord SecureWire 500 and SafeWord SecureWire 2500 Identity and Access Management Appliances are multi-chip standalone cryptographic modules. SecureWire 500 and SecureWire 2500 are enclosed in a hard, opaque metal case that completely encloses all of the internal components of the modules. There are only a limited set of vent holes provided in the case, and these are all obscured to prevent viewing of the internal components of the module. Tamper-evident labels are applied to the case of the 2500 and 500 to provide physical evidence of attempts to remove the case or hard drives of the modules. All of the modules' components are production grade.

The 2500 and 500 were tested and found conformant to the EMI/EMC requirements specified by 47 Code of Federal Regulations, Part 15, Subpart B, Unintentional Radiators, Digital Devices, Class A (i.e., for business use).

## 1.5 Operational Environment

The operational environment requirements do not apply to the 2500 and 500. The modules do not provide a general purpose operating system and only allow firmware updating using digital signed Secure Computing firmware updates. Additionally, only upgrades validated to FIPS 140-2 may be activated, as described in the Crypto-Officer Management guidance below.

## 1.6 Cryptographic Key Management

SecureWire 2500 and 500 implement FIPS-approved algorithms in both hardware and software. The following is a list of the supported algorithms which are Approved or allowed under FIPS 140-2.

**Digital signatures**

- DSA - FIPS 186-2 (certificate #129, 130, 131)

- RSA - PKCS#1 (certificate #55, 56)

**DSA key generation**

- DSA - FIPS 186-2 (certificate #131)

**RSA key generation**

- RSA key generation – Appendix B.4 of ANSI X9.31 (certificate #55, 56)

**Symmetric encryption**

- DES (CBC mode – for legacy use only – transitional phase only – valid until May 19, 2007) – FIPS 46-3 (certificate #299, 300, 301, 302, 303, 304)
- TripleDES (ECB,CBC mode) – FIPS 46-3 (certificate #319, 320, 321, 322, 323, 325, 326)
- AES (ECB, CBC mode) – FIPS 917 (certificate #229, 230, 231, 232, 233, 234, 235)

**Hashing**

- SHA-1 – FIPS 180-2 (certificate #308, 309, 310, 311, 312, 313, 314)

**MAC'ing**

- HMAC with SHA-1 – FIPS 198 (certificate #41, 42¸ 43, 44, 45; SHA- certificate #308, 309, 310, 311, 314 respectively)

**Random number generation**

- X9.31 PRNG – Appendix A.2.4 of ANSI X9.31 (certificate #69, 70, 71, 72, 73, 74)

**Key Establishment Methodology**

- X9.31 PRNG – Appendix A.2.4 of ANSI X9.31 (certificate #69, 70, 71, 72, 73, 74)
- Diffie-Hellman (key agreement, key establishment methodology provides 70, 80, or 96-bits of encryption strength)[1]
- RSA (key wrapping, key establishment methodology provides between 80 and 110-bits of encryption strength)

Additionally, the module implements the following non-FIPS-approved algorithms for use in a FIPS mode of operation.

- Hardware RNG

The module also implements the following non-FIPS-approved algorithms, which are not used in a FIPS mode of operation.

- MD5
- MD5 HMAC
- RC4

[1] In order to operate in an Approved mode of operation compliant to FIPS 140-2, Diffie-Hellman keys of 1024-bits and larger must be used.

The module supports the following critical security parameters:

**Table 9 - List of Cryptographic Keys, Cryptographic Key Components, and CSPs**

| Key | Key Type | Generation / Input | Storage | Zeroization | Use |
|---|---|---|---|---|---|
| RSA Public Keys | RSA (1024-2048 bit) | RSA public key generated internally by the module using X9.31 key generation; or externally generated loaded onto the module in a certificate (during IKE, TLS, or SSH handshakes) and/or over a management TLS session | Stored on the hard drive in database in plaintext; or not stored - in volatile memory only | Zeroized by deleting the key through the SWCC API. Zeroized during execution of the zeroization command. | Used during TLS, SSH, or IPSec session establishment. Used for certificate verification. |
| RSA Private Keys | RSA (1024-2048 bit) | RSA private key generated internally by the module using X9.31 key generation; or externally generated loaded onto the module over a management TLS session | Stored on the hard drive in database in plaintext | Zeroized by deleting the key through the SWCC API. Zeroized during execution of the zeroization command. | Used during TLS, SSH, or IPSec session establishment. |
| DSA Public Keys | DSA (1024 bit) | DSA public key generated internally by the module; or externally generated loaded onto the module in a certificate (during IKE or SSH handshakes) and/or over a management TLS session | Stored on the hard drive in database in plaintext; or not stored - in volatile memory only | Zeroized by deleting the key through the SWCC API. Zeroized during execution of the zeroization command. | Used during SSH or IPSec session establishment. Used for certificate verification. |
| DSA Private Keys | DSA (1024 bit) | DSA private key generated internally by the module; or externally generated loaded onto the module over a management TLS session | Stored on the hard drive in database in plaintext | Zeroized by deleting the key through the SWCC API. Zeroized during execution of the zeroization command. | Used during SSH or IPSec session establishment. |
| HA shared secret key | AES (128 bits) | Externally generated loaded onto the module over a management TLS session | Stored on the hard drive in database in plaintext | Zeroized by deleting the key through the SWCC API. Zeroized during execution of the zeroization command. | Used during High Availability synchronization |

| Key | Key Type | Generation / Input | Storage | Zeroization | Use |
|---|---|---|---|---|---|
| TLS session keys | AES (128, 256 bits), Triple-DES (168 bits), DES (56 bits), SHA-1 HMAC (160 bits) | Negotiated during TLS session establishment. | Not stored - in volatile memory only in plaintext. | Zeroized when the module is powered down. | Used to encrypt/MAC the TLS session. |
| IPSec session keys | AES (128, 192, 256 bits), Triple-DES (168 bits), DES (56 bits), SHA-1 HMAC (160 bits) | Negotiated during IPSec session establishment; or externally generated loaded onto the module over a management TLS session | Stored on the hard drive in database in plaintext; or not stored - in volatile memory only | Zeroized when the module is powered down; or Zeroized by deleting the key through the SWCC API. Zeroized during execution of the zeroization command. | Used to encrypt/MAC the IPSec session. |
| SSH session keys | AES (128, 256 bits), Triple-DES (168 bits), SHA-1 HMAC (160 bits) | Negotiated during SSH session establishment. | Not stored - in volatile memory only in plaintext. | Zeroized when the module is powered down. | Used to encrypt/MAC the SSH session. |
| Diffie-Hellman key pairs | Diffie-Hellman (768, 1024, or 1536 bit) | Generated for IKE/IPSec and SSH session establishment using an X9.31 PRNG. | Not stored - in volatile memory only in plaintext. | Zeroized when the module is powered down. | Used by the module in establishing a session key during IKE/IPSec and SSH negotiation. |
| Operator passwords | Passwords | Entered into module by remotely authenticating operator over an IPSec, TLS, or SSH session or locally over directly connected serial port, verified internally | Stored on the hard drive in database in plaintext; or not stored - in volatile memory only | Zeroized when the password is updated with a new one or the user is deleted. Zeroized during execution of the zeroization command. | Used for authentication of Crypto-Officer and User passwords. |
| X9.31 seeds | Seed | Generated internally by querying the Cavium hardware RNG, or generated using entropy gathering routines | Not stored - in volatile memory only in plaintext. | Zeroized when the module is powered down. | Used to seed the X9.31 PRNGs |

## 1.7 Self-Tests

SecureWire 2500 and 500 perform a series of self-tests to verify the correct operation of the modules. These tests are both perform at power-up and continuously during normal operations upon certain conditions. The following is a list of self-tests performed by the modules.

**Power-up Self-tests**

- Software Integrity Test – The modules verify that their firmware has not been modified by verifying CRC-32 checksums over their firmware.
- Cryptographic Algorithm Tests – The 2500 and 500 perform cryptographic algorithm tests on all FIPS-approved algorithms at power-up to verify the correct operation of the algorithm implementations.
    - DES Known Answer Tests (KATs)
    - Triple-DES-ECB KATs
    - AES-CBC KATs
    - SHA-1 KATs
    - HMAC with SHA-1 KATs
    - DSA Pair-wise Consistency Tests (sign/verify)
    - RSA Pair-wise Consistency Tests (sign/verify)
    - RSA Pair-wise Consistency Tests (encrypt/decrypt)
    - X9.31 PRNG KATs

**Conditional Self-tests**

- Continuous Random Number Generator Tests – This test is run upon generation of random data by the module's random number generators to detect failure to a constant value.
- Software update test – This test is run upon updating of the module's firmware. A digital signature is verified over the firmware update to ensure the integrity of the firmware update.
- RSA Pair-wise Consistency Tests (sign/verify) – This test is run upon generation of a new RSA key pair to verify the correct operation of the newly generated key pair.
- RSA Pair-wise Consistency Tests (encrypt/decrypt) – This test is run upon generation of a new RSA key pair to verify the correct operation of the newly generated key pair.
- DSA Pair-wise Consistency Tests (sign/verify) – This test is run upon generation of a new DSA key pair to verify the correct operation of the newly generated key pair.

If any of the power-up self-tests fail, the modules enter an error state and output an error indicator. If any of the conditional self-tests fails, the service that caused the failure enters an error state, outputs an error indicator, and terminates.

## 1.8 Design Assurance

AccuRev/CM is used for SecureWire's configuration management of source code and related documentation and Arena Solutions PLM for its configuration management of hardware components and related documentation. Both systems provide access control, versioning, and logging.

Additionally, Microsoft Visual Source Safe (VSS) version 6.0 is used to provide configuration management for the SafeWord SecureWire 500 and SafeWord SecureWire 2500 Identity and Access Management Appliances's FIPS documentation. This software provides access control, versioning, and logging.

## 1.9 Mitigation of Other Attacks

This section is not applicable.

# 2 Secure Operation

The SafeWord SecureWire 500 and SafeWord SecureWire 2500 Identity and Access Management Appliances meets Level 2 requirements for FIPS 140-2. The sections below describe how to place and keep the module in FIPS-approved mode of operation.

## 2.1 Crypto-Officer Guidance

### 2.1.1 Initial Setup

The 500 and 2500 are available directly from Secure Computing through shipping using a bonded carrier or delivery directly, and by direct pickup from a Secure Computing facility. The Crypto-Officer is responsible for inspecting the module and its packing upon receipt for signs of tamper.

The 500 or 2500 is provided in a Secure Computing box sealed with tape. Inside of this box, the 2500 or 500 is sealed with tape in an anti-static bag. The Crypto-Officer must inspect the box, packing materials, and module for signs of tamper, including damage to the box, packing materials, or the module itself. If tamper-evidence is found, the Crypto-Officer should contact Secure Computing immediately and not use the module.

If the 500 or 2500 have been shipped without tamper-evident labels applied, then the Crypto-Officer must apply these labels.

The following steps detail application of the labels for the 500.

1. Ensure the system is unplugged.
2. Clean the areas to which the tamper-evident labels will be applied to remove any grease, dirt, etc. Rubbing alcohol can be used for this purpose.
3. Apply a tamper-evident label at the seam across the bottom center of the faceplate and bottom of the chassis of the front of the module.

**Figure 5 – Apply label to bottom center of the faceplate of the 500**

4. Apply a tamper-evident label at the seam across the top center of the faceplate and top of the chassis of the front of the module.

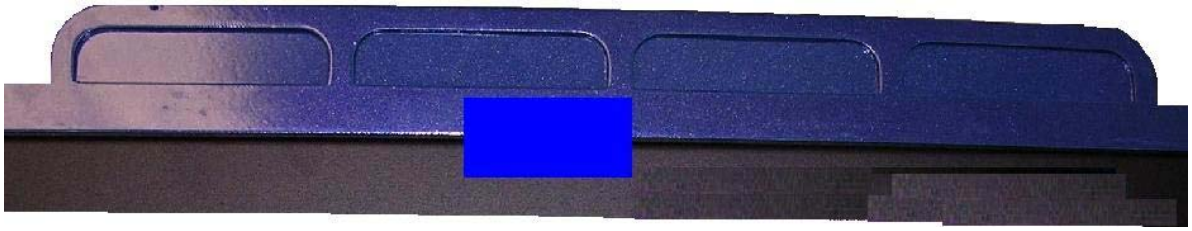

**Figure 6 – Apply label to top center of the faceplate of the 500**

5. Apply a tamper-evident label across the top of the chassis and the hard drive bay of the rear of the module.


**Figure 7 – Apply label to upper hard drive bay of the rear of the 500**

6. Log the serial numbers of the applied labels.

7. Allow a minimum of 24 hours for the labels to cure.

The following steps detail application of the labels for the 2500.

1. Ensure the system is unplugged.

2. Clean the areas to which the tamper-evident labels will be applied to remove any grease, dirt, etc. Rubbing alcohol can be used for this purpose.

3. Apply a tamper-evident label at the seam across the bottom center of the faceplate and bottom of the chassis of the front of the module.

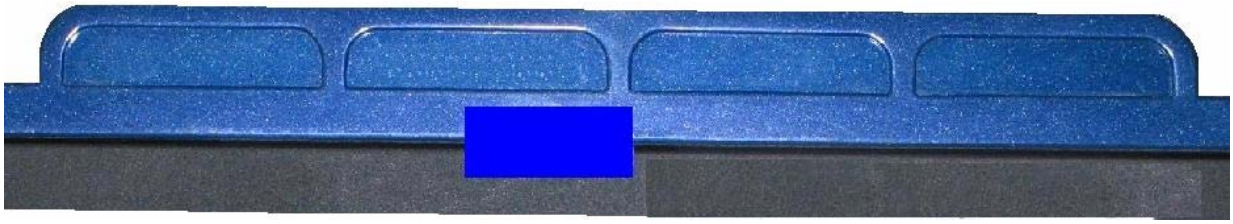

**Figure 8 – Apply label to bottom center of the faceplate of the 2500**

4. Apply a tamper-evident label at the seam across the top center of the faceplate and top of the chassis of the front of the module.

**Figure 9 – Apply label to top center of faceplate of the 2500**

5. Apply a tamper-evident label across the top of the chassis and the upper hard drive bay of the rear of the module.

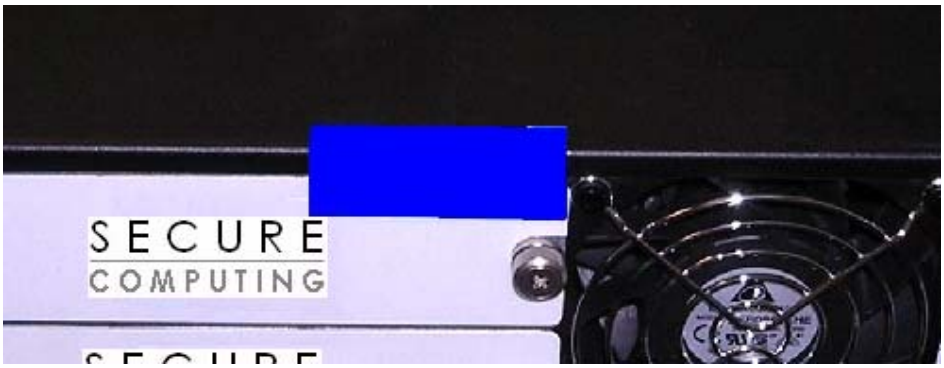


**Figure 10 – Apply label to upper hard drive bay of the rear of the 2500**

6. Apply a tamper-evident label across the bottom of the chassis and the lower hard drive bay of the rear of the module.

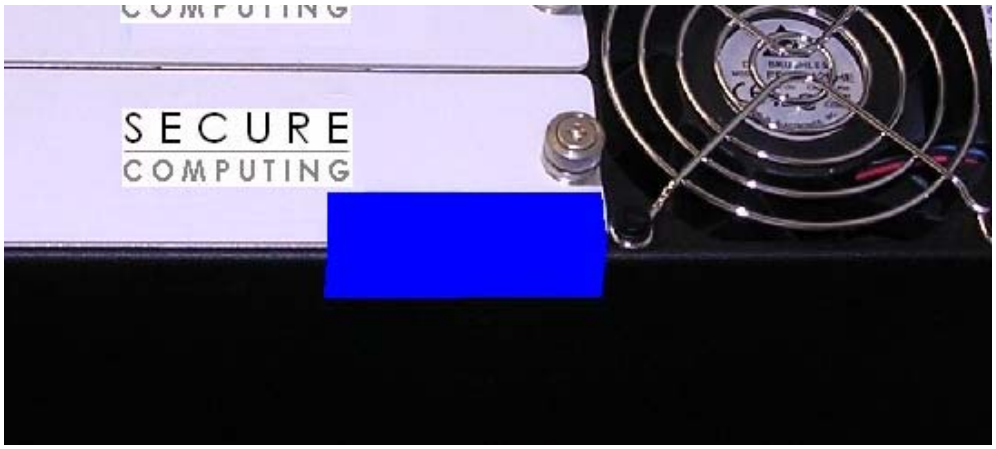


**Figure 11 – Apply label to lower hard drive bay of the rear of the 2500**

7. Log the serial numbers of the applied labels.

8. Allow a minimum of 24 hours for the labels to cure.

### 2.1.2 Initialization

The module ships with a default Crypto-Officer account (“admin”) and a default password for this account (“caymas”). The Crypto-Officer must change this password and must use a password a minimum of 6 characters in length. This can be performed through the Administrators->Administrators screen in SWCC GUI. Once completed, the Crypto-Officer must apply the changes.

Next, the Crypto-Officer must import or create a new key pair to replace the default SecureWire key pair that is shipped with the module, and this key pair must be a minimum of 1024 bits for RSA. This can be done through the System->Certificates screen in SWCC GUI. Once completed, the Crypto-Officer must apply the changes.

After creating a new key pair, the Crypto-Officer must configure a new SSL server to replace the default SecureWire SSL server, and this server must be configured with a certificate other than the default SecureWire

certificate. Only FIPS-approved cipher suites may be configured for this SSL server, and these cipher suites are DES SHA-1 – for legacy use only, 3DES SHA-1, AES128 SHA-1, and AES256 SHA-1. This can be done through the System->SSL Servers screen in SWCC. Once completed, the Crypto-Officer must apply the changes. (Note: While the term SSL is used here, once configuration for FIPS is completed, the actual protocol used will be TLSv1 and not SSL.)

Once these steps have been completed, the Crypto-Officer can proceed with removing the default SSL server and the default SecureWire key pair. The Crypto-Officer must be careful to first remove the default SSL server, and then to remove the default SecureWire key pair. Removing the default SSL server can be accomplished through the SWCC GUI on the System->SSL Servers screen by selecting the default SSL server and selecting to remove it. Removing the default SecureWire key pair can be accomplished through the SWCC GUI on the System->Certificates screen by selecting the default SecureWire certificate and selecting to remove it. Once completed, the Crypto-Officer must apply the changes.

The Crypto-Officer must enable all of the FIPS flags in order to enable or disable certain functionality provided by the module for FIPS mode. The flags can be easily configured on the System->Settings->FIPS Settings screen in SWCC. The following flags must be checked.

- Enable SWCC Management over SecureWire Tunnel
- Restrict Front-End To TLS
- Restrict Back-End to TLS and FIPS Ciphers
- Verify Software Image Integrity on Startup
- Restrict SWCC and CLI to One Administrator
- Disable Debug Shell Sessions
- Encrypt CSPs for High-Availability Communication
- Allow Only Encrypted Traffic
- Restrict SWCC Management to SecureWire Tunnel
- Enable Cryptographic Module Self-Test
- Zeroize Upon Restore to Factory Defaults

These flags ensure that only TLS is used with FIPS-approved cipher suites, disable SSL, enable the FIPS self-tests, disable multiple administrators using the same management interfaces, disable shell access, ensure that all CSPs exiting the module are encrypted, disable any bypass capabilities, enable full zeroization when switching to a factory default state (i.e., out of the FIPS configuration), and enforce an encrypted management sessions. Once completed, the Crypto-Officer must apply the changes.

High availability, when configured, creates a channel that will encrypt CSPs exchange between the two modules established in the high availability configuration. An AES key must be entered into the module for this purpose, and the Crypto-Officer is responsible for entering a 128 bit AES key when using high availability.

The Crypto-Officer must disable port 80 on the module. This flag can be configured on the System->Settings->Basic Settings screen in SWCC. The “Disable Port 80” flag must be checked. This disables plaintext HTTP sessions through the module.

The Crypto-Officer must configure an Authentication Policy for passwords that requires password to be a minimum of 6 characters in length, and all pre-shared keys must be a minimum of 16 characters in length. By default, Crypto-Officer passwords must be a minimum of 6 characters in length. Additionally, the Crypto-Officer must take care to choose secure passwords containing upper case letters, lower case letters, numbers, and symbols.

At this point, the module must be rebooted to enable all of the changes. Upon reboot, initialization of the module for FIPS is complete and the module is now configured securely.

Note: While outside of the module's boundary, the Crypto-Officer must configure their external software (e.g., a web browser) to use TLS when attempting to interface with the module using SSL/TLS.

### 2.1.3 Management

The Crypto-Officer must be sure to only configure cryptographic services for the module using the FIPS-approved algorithms, as listed Cryptographic Key Management section above. TLSv1 and IPSec must only be configured to use FIPS-approved cipher suites, and only digital certificates generated with FIPS-approved algorithms may be utilized. RSA key pairs must be a minimum of 1024 bits in length, and DSA key pairs must be equal to 1024 bits in length. DES must only be used for legacy purposes and is not to be used for management connections to the module.

The Crypto-Officer must configure all traffic flowing through the module to undergo cryptographic processing. All policies must require that services are accessed over encrypted session.

All IPSec Site to Site Policies must be configured prior to defining the network-based Resource Access Rules (RARs) that utilize the IPSec Site to Site Policies. When any IPSec Site to Site Policy is deleted, the module must be rebooted.

If used, third party authentication mechanism must be configured to follow the password guidelines set out in the previous paragraph. Active directory and LDAP authentication servers must be configured to use a TLSv1 connection with the module. RADIUS servers must have a minimum of a 6 character shared secret/password in place with the module, and SecureID must be configured to use DES.

The Crypto-Officer must not configure FTP transfer of coredumps, as these may contain sensitive information. Additionally, FTP transfers of files are not permitted, and only SCP may be used, which pipes data over an SSH connection. The module is configured by default to only use FIPS-approved cipher suites with SSH and SCP, and this must not be modified.

The module permits upgrades through the CLI using the release upgrade command. When upgrading, the module verifies a DSA digital signature over the upgrade to ensure that it is an unmodified, SecureWire firmware update. However, the Crypto-Officer must also ensure that the upgrade is validated to FIPS 140-2 by checking the version of the firmware and verifying this has been validated to FIPS 140-2 before activating the upgrade. Since upgrading the module with a release that has not been validated to FIPS 140-2 will take the module out of FIPS mode, the Crypto-Officer must either zeroize the module (see Zeroization below) before activating the upgrade or not proceed with activating the upgrade.

The Crypto-Officer should periodically backup the configuration of the module.

The Crypto-Officer must periodically check the module for signs of tamper-evidence, including unusual dents, scrapes, or damage to tamper-evident labels, and verify the tamper-evident labels still have the proper serial numbers. Additionally, the Crypto-Officer should monitor logs and alerts for the module for strange activity. If indications of suspicious activity are found, the Crypto-Officer should immediately take the module offline and investigate.

### 2.1.4 Zeroization

At the end of its life cycle or when taking the module out of FIPS mode, the module must be fully zeroized to protect CSPs. This can be accomplished through the SWCC GUI on the System->Settings->FIPS Settings screen or by resetting to a factory default through the SDM. The Crypto-Officer must wait until the module has successfully rebooted in order to verify that zeroization has completed.

## 2.2 User Guidance

The User does not have the ability to configure sensitive information on the module, with the exception of their password. The User must be diligent to pick strong passwords (8 characters or greater, a minimum of alphanumeric), and must not reveal their password to anyone. Additionally, the User should be careful to protect any secret/private keys in their possession, such as IPSec session keys.

Note: While outside of the module's boundary, the User must be sure configure their external software (e.g., a web browser) to use TLS when attempting to interface with the module using SSL/TLS.

## 2.3 Network User Guidance

The Network User does not have the ability to configure sensitive information on the module.

# 3 Acronyms

**Table 10 - Acronyms**

| Acronym | Definition |
|---|---|
| ANSI | American National Standards Institute |
| API | Application Programming Interface |
| CLI | Command Line Interface |
| CMVP | Cryptographic Module Validation Program |
| CRC | Cyclical Redundancy Check |
| CSE | Communications Security Establishment |
| CSP | Critical Security Parameter |
| DSA | Digital Signature Algorithm |
| EMC | Electromagnetic Compatibility |
| EMI | Electromagnetic Interference |
| FCC | Federal Communication Commission |
| FIPS | Federal Information Processing Standard |
| GUI | Graphical User Interface |
| HA | High Availability |
| HMAC | (Keyed-) Hash MAC |
| IKE | Internet Key Exchange |
| IP | Internet Protocol |
| IPSec | IP Security |
| KAT | Known Answer Test |
| LAN | Local Area Network |
| LED | Light Emitting Diode |
| MAC | Message Authentication Code |
| NIST | National Institute of Standards and Technology |
| NVLAP | National Voluntary Laboratory Accreditation Program |
| RAM | Random Access Memory |
| RAR | Resource Access Rule |
| RNG | Random Number Generator |
| RSA | Rivest Shamir and Adleman |
| SCP | Secure Copy Protocol |
| SDM | SecureWire Display Module |
| SHA | Secure Hash Algorithm |

| Acronym | Definition |
|---|---|
| SSH | Secure SHell |
| SSL | Secure Socket Layer |
| SWCC | SecureWire Control Center |
| TLS | Transport Layer Security |