TIBCO LogLogic®, Inc

LogLogic Communications Cryptographic Module

Software Version: 1.0

FIPS 140-2 Non-Proprietary Security Policy

FIPS Security Level: 1
Document Version: 1.1

Prepared for:

TIBCO
The Power of Now™

TIBCO LogLogic®, Inc
110 Rose Orchard Way, Suite 200
San Jose, California 95134
Unites States of America
Phone: +1 (888) 347-3883

Email: info@loglogic.com
http://www.loglogic.com

Prepared by:

Corsec®

Corsec Security, Inc.
13135 Lee Jackson Memorial Highway, Suite 220
Fairfax, VA 22033
Unites States of America
Phone: +1 (703) 267-6050

Email: info@corsec.com
http://www.corsec.com

Table of Contents

Table of Figures

List of Tables

1 Introduction

1.1 Purpose

This is a non-proprietary Cryptographic Module Security Policy for the LogLogic Communications Cryptographic Module from TIBCO LogLogic, Inc This Security Policy describes how the LogLogic Communications Cryptographic Module meets the security requirements of Federal Information Processing Standards (FIPS) Publication 140-2, which details the U.S. and Canadian 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) and the Communications Security Establishment Canada (CSEC) Cryptographic Module Validation Program (CMVP) website at http://csrc.nist.gov/groups/STM/cmvp.

This document also describes how to run the module in a secure FIPS-Approved mode of operation. This policy was prepared as part of the Level 1 FIPS 140-2 validation of the module. The LogLogic Communications Cryptographic Module is referred to in this document by name, as crypto-module, or the module.

1.2 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 LogLogic website (http://www.loglogic.com) contains information on the full line of products from TIBCO LogLogic.
- The CMVP website (http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm) contains contact information for individuals to answer technical or sales-related questions for the module.

1.3 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 Model document
- Other supporting documentation as additional references

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

2 Communications Cryptographic Module

2.1 Overview

TIBCO LogLogic, Inc is an Information Technology (IT) Data Management company, providing enterprise-class log management infrastructure and analysis to empower a customer to turn raw IT data into immediate business intelligence.

Their goal is to provide 360^o insight, allowing for greater control in compliance, security, and efficiency within the customer's IT infrastructure. To accomplish this, TIBCO LogLogic has developed several appliances that specialize in handling high-volume log data, for use in Enterprise-level IT management. The appliances themselves act as log collectors, log storage units, and log management appliances. These are the functions of the LX, ST, and MX hardware designations, respectively.

- LX-Series:
  - Provides real-time log data collection and analysis for scalable business deployments
- ST-Series:
  - Automates log archive and retrieval processes
- MX-Series:
  - Provides an all-in-one solution, pairing real-time log data collection and analysis with longer data retention, for a broad-based solution.

The LogLogic Communications Cryptographic Module is a tunneling application, developed by LogLogic and deployed within their hardware appliances and virtual appliances.

The purpose of the LogLogic Communications Cryptographic Module is to properly establish a secure, encrypted tunnel between LogLogic appliances for the secure transmission of log data. The module is a 32-bit Linux executable (.elf), called ll_tunnel, providing general-purpose multi-algorithm cryptographic services. The purpose of the module is to provide a single module for cryptographic functionality that can establish a secure tunnel and provide availability of FIPS-Approved algorithms for use in the secure tunnel.

The LogLogic Communications Cryptographic Module is validated at the FIPS 140-2 Section levels listed in Table 1.

Table 1 – Security Level Per FIPS 140-2 Section

Section	Section Title	Level
1	Cryptographic Module Specification	1
2	Cryptographic Module Ports and Interfaces	1
3	Roles, Services, and Authentication	1
4	Finite State Model	1
5	Physical Security	N/A[1]
6	Operational Environment	1
7	Cryptographic Key Management	1
8	EMI/EMC[2]	1

[1] N/A – Not Applicable

[2] EMI/EMC – Electromagnetic Interference / Electromagnetic Compatibility

Section	Section Title	Level
9	Self-tests	1
10	Design Assurance	1
11	Mitigation of Other Attacks	N/A
14	Cryptographic Module Security Policy	1

2.2 Module Specification

The LogLogic Communications Cryptographic Module is a software module with a multi-chip standalone embodiment. The overall security level of the module is 1. The LogLogic Communications Cryptographic Module is in the form of a software executable running on Oracle Enterprise Linux (OEL). The physical cryptographic boundary of the module is the hardware platform that it runs on. The module resides on the hard-drive of the hardware platform. The module is loaded and executes in volatile RAM.

The module is called by engine_stunnel, which passes configuration data concerning its cryptographic operation. The module then uses this data to create a secure tunnel. The configuration data consists of a localport, the Internet Protocol (IP) of the host and a targetport. The localport is a TCP[3]/UDP[4] port that defines where the module will listen for incoming data, in the client configuration. The IP designates the hostname to send data to. The targetport is a TCP/UDP port that will define where the data is being sent to. Once this is completed, the module listens for incoming traffic, which may come from either the secure tunnel or the forwarder.

The forwarder is an application, running on the appliance, which governs data to be sent through the secure tunnel. The forwarder passes information to the localport, which sends it into the module. The module then performs cryptographic functions on this data, to encrypt it, and then sends it out via the tunnel.

The collector is an application, running on the appliance, which receives data sent through the secure tunnel. The module receives data via the secure tunnel, performs cryptographic operations to decrypt it, then sends this data on through the targetport and into the collector.

The module can run in either a client or server configuration. Figure 1 and Figure 2 illustrate the module in these configurations, executing in memory. The module boundary in the figures is indicated by a dotted red line. The configuration only determines where information flows to and from the module. The configuration does not have bearing on cryptographic operation.

[3] TCP - Transmission Control Protocol
[4] UDP – User Datagram Protocol

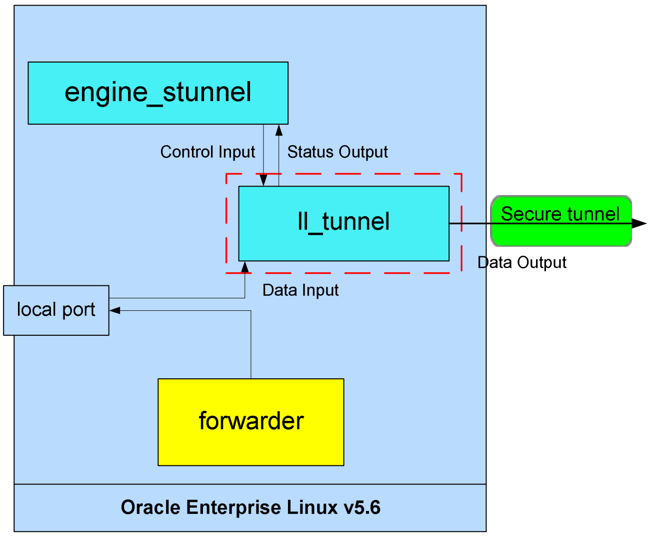

Figure 1 – FIPS 140-2 Logical Block Diagram (ll_tunnel in client configuration)

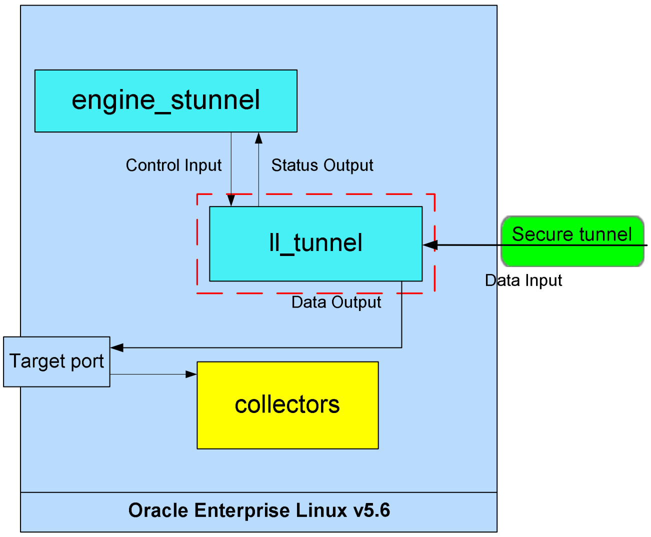

Figure 2 – FIPS 140-2 Logical Block Diagram (ll_tunnel in server configuration)

The cryptographic module was tested and found compliant on the following platforms:

- LX 820, 1020, 4020
- ST 1020, 2020-SAN, 4020
- MX 3020

These platforms represent the physical boundary for the module. The hardware block diagrams for these platforms are listed below in Figure 3 and Figure 4.

The LX1020, LX4020, MX3020, ST1020, and ST2020-SAN are designed around the block diagram shown in Figure 4. The LX 820, LX 1020, ST 1020, and MX 3020 appliances employ a single-CPU[5] quad-core

[5] CPU - Central Processing Unit

design. The LX4020, ST 2020-SAN, and ST 4020 appliances employ a dual-CPU quad-core design. The code execution of the module does not change between the two different processor configurations. Note that the dotted line surrounding the physical components represents the module's physical cryptographic boundary.

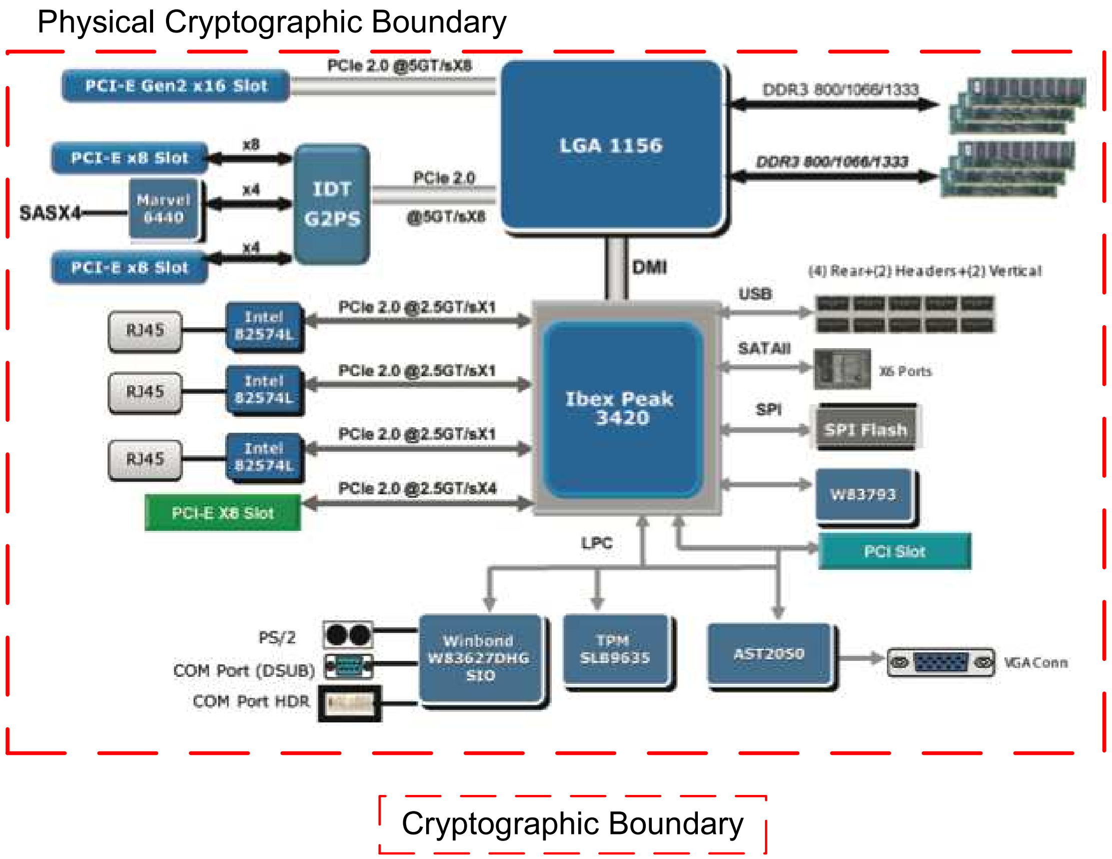

Figure 3 – LX 820 Block Diagram

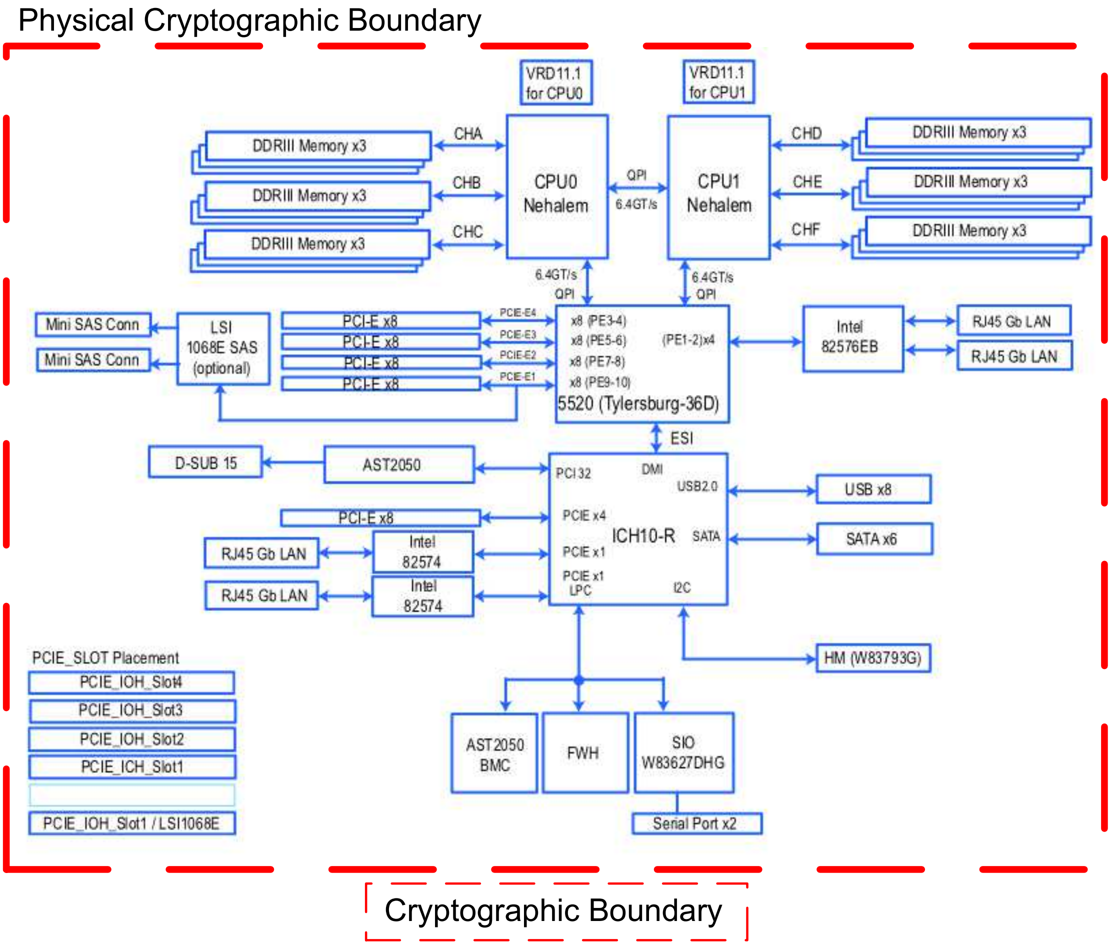

Figure 4 – LX 1020/ST 1020/ST 2020-SAN/MX 3020/LX 4020/ST 4020 Block Diagram

Note that, although the figure above shows a dual-CPU appliance block diagram, the single-CPU appliance block diagram is identical except for number of CPUs.

2.3 Module Interfaces

All interfaces of the module can be described through a series of Command-Line (CLI) commands to the module and their associated returns from the module. Since the module is defined as software, the only interfaces that exist are those between the module and the other software and Operating System (OS) processes that communicate with it.

All of these physical interfaces are separated into logical interfaces defined by FIPS 140-2, as described in Table 2.

Table 2 – FIPS 140-2 Logical Interface Mappings

FIPS 140-2 Logical Interface	Physical Port/Interface	Communications Cryptographic Module Port/Interface
Data Input	USB, Ethernet	Parameters for a command that takes the data to be used or processed by the module
Data Output	USB, Ethernet	Parameters for a command that specify where the result of the function is stored
Control Input	Alarm Reset, Power Button, Serial port, Ethernet	Command parameters used to control the operation of the module
Status Output	LED, Ethernet, Serial Port	Return values for commands or parameters
Power	Power Port	Not Applicable

2.4 Roles and Services

There are two roles in the module (as required by FIPS 140-2) that operators may assume: a Crypto Officer role and a User role.

2.4.1 Crypto Officer Role

The Crypto Officer role has the ability to set the type of compression and encryption of data through the module. These settings will apply to all users. Descriptions of the services available to the Crypto Officer role are provided in the table below. Please note that the keys and CSPs listed in the table indicate the type of access required using the following notation:

- R – Read: The CSP is read.
- W – Write: The CSP is established, generated, modified, or zeroized.
- X – Execute: The CSP is used within an Approved or Allowed security function or authentication mechanism.

Table 3 – Crypto Officer Services

Service	Description	CSP and Type of Access
Start module	Loads the module into memory and starts it	N/A
Stop module	Stops module operations and unloads it from memory	N/A
Zeroize Keys	Zeroizes all keys and CSPs created by the module	All – W
Check FIPS mode	Checks to see whether the module is operating in FIPS mode	N/A
Run Self-Tests	Runs On-Demand Self-Tests for cryptographic algorithms	All – WRX
Process Cleanup	Engine_stunnel will kill timed-out (20s) instances of ll_tunnel	All – W

2.4.2 User Role

The User role has the ability to perform communication of log data and establishment/manipulation of client-server connection. Descriptions of the services available to the User role are provided in the table below.

Table 4 – User Services

Service	Description	CSP and Type of Access
Encryption	The module performs encryption operations on information from the Data Input interface	AES – X
Decryption	The module performs decryption operations on information from the Data Input interface	AES – X
Key Establishment	Diffie-Hellman method of performing key establishment for transmission of session key	Diffie-Hellman – RWX
Run Self-Tests	Runs On-Demand Self-Tests for cryptographic algorithms	All – RWX
HMAC SHA-1 Operation	Generate HMAC SHA-1 Hash	N/A
SHA-1 Operation	Generate a SHA-1 hash	AES – W Diffie-Hellman – R
SHA-256 Operation	Generate a SHA-256 hash	N/A
Random Number Generation	Generate a random value for the Diffie-Hellman nonce from system entropy	Diffie-Hellman – W

2.5 Physical Security

The LogLogic Communications Cryptographic Module is a software module, which FIPS defines as a multi-chip standalone cryptographic module. As such, it does not include physical security mechanisms. As such, the FIPS 140-2 requirements for physical security are not applicable.

2.6 Operational Environment

The module has been validated to be deployed on the hardware appliances defined in Section 2.2. The validated LogLogic appliances are deployed with Oracle Enterprise Linux (OEL) v5.6. The Crypto-Officer shall ensure that the OEL OS is configured to a single user mode of operation. This operational environment is modifiable, however the OS protects the memory space that the module runs in, securing its CSPs against tampering or disclosure. The calling application that loads the module is the only operator. The Operating System is responsible for maintaining separation of the different operators and their access to processes running on the OS.

LogLogic affirms that the virtual machine version of this module is binary-compatible with the module deployed on LogLogic hardware. All algorithms will operate identically on the virtual machine. The virtual machine implementation of the module will be configured to run on a general-purpose computer, running with an Intel Core 2 Quad Processor (Q9550). The virtual environment will be running on an LogLogic's Enterprise Virtual Appliance (EVA) within VMWare ESXi, running OEL v5.6, and VMWare

Workstation, running Windows 7.

The module also executes an HMAC SHA-1 integrity test on itself, at runtime, to ensure that it has not been compromised.

2.7 Cryptographic Key Management

The module implements the FIPS-Approved algorithms listed in Table 5 below.

Table 5 – FIPS-Approved Algorithm Implementations

Algorithm	Certificate Number
AES-CFB – 256-bit key size	#1926
SHA-1, SHA-256	#1691
HMAC SHA-1	#1160
ANSI X9.31 Pseudo-Random Number Generator	#1013

Additional information concerning ANSI X9.31 and specific guidance on transitions to the use of stronger cryptographic keys and more robust algorithms is contained in NIST Special Publication 800-131A.

Additionally, the module utilizes the following non-FIPS-Approved algorithm implementation:

- Diffie-Hellman – for key establishment (2048-bit key establishment methodology provides 112 bits of encryption strength)

The module supports the critical security parameters (CSPs) listed below in Table 6.

Table 6 – List of Cryptographic Keys, Cryptographic Key Components, and CSPs

Key	Key Type	Generation / Input	Output	Storage	Zeroization	Use
D-H Public Key	2048-bit D-H Key	• Client-side: Loaded from external file • Server-side: Generated during key negotiation	• Client-side: Exported in plaintext via Data Output interface • Server-side: Exported in plaintext via Data Output interface	• Client-side: Plaintext in non-volatile memory • Server-side: Plaintext in volatile memory	• Client-side: Zeroize Keys service from CO Services table • Server-side: reboot module/terminate session	Exchanging shared Session Key during key negotiation
D-H Private Key	2048-bit D-H key	• Client-side: Loaded from external file • Server-side: Generated during key negotiation	• Never exit the module	Client-side: plaintext in non-volatile memory Server-side: plaintext in volatile memory	• Client-side: Zeroize Keys service from CO Services table • Server-side: reboot module/terminate session	Exchanging shared Session Key during key negotiation
Session Key	AES-CFB 256-bit key	Generated during key negotiation	Never exit the module	Plaintext in volatile memory	Reboot, session termination	Encrypt or decrypt data
Software Integrity Key	HMAC SHA-1 key	Never	Never exit the module	Within module as plaintext file	N/A	Used to perform the software integrity test on module load
PRNG Seed	PRNG Seed 128-bit value	Imported from /dev/random	Never exit the module	Plaintext in volatile memory	Reboot, session termination	Entered into PRNG to generate Diffie-Hellman key
PRNG Seed Key	Seed key (256-bit)	Imported from /dev/random	Never exit the module	Plaintext in volatile memory	Reboot, session termination	Entered into PRNG to generate Diffie-Hellman key

2.8 Self-Tests

2.8.1 Power-Up Self-Tests

The LogLogic Communications Cryptographic Module performs the following self-tests at power-up:

- Software integrity check using HMAC SHA-1
- Known Answer Tests (KATs)
    - AES KAT
    - SHA-1 KAT
    - SHA-256 KAT
    - ANSI X9.31 PRNG KAT

2.8.2 Conditional Self-Tests

The LogLogic Communications Cryptographic Module performs the following conditional self-tests:

- Continuous RNG test for ANSI X9.31 PRNG

2.9 Mitigation of Other Attacks

This section is not applicable. The modules do not claim to mitigate any attacks beyond the FIPS 140-2 Level 1 requirements for this validation.

3 Secure Operation

The LogLogic Communications Cryptographic Module meets Level 1 requirements for FIPS 140-2. The sections below describe how to place and keep the module in FIPS-approved mode of operation.

The cryptographic module implements a software integrity test that consists of an HMAC SHA-1 hash computed over the module executable. During the power-up self-tests phase, the hash is compared with the stored Communication Cryptographic Module instance. If the computed hash value matches the stored hash, then the test is passed. Otherwise, the test is failed and the module enters an error state where no cryptographic functionality is allowed.

3.1 Initial Setup

The module is installed on the LogLogic hardware appliances by LogLogic. Deployment of the virtual machine requires that Enterprise Virtual Appliance be installed on the host GPC. The LogLogic Communications Cryptographic Module is installed as a component of the Enterprise Virtual Appliance.

The Crypto-Officer must change several files in order for the module to be initialized in FIPS mode. First, they must change the compatibility.xml file. There is a property labeled "ID_length" that must be set to "256". This defines the bit-size that will be used for the AES algorithm. Within dhparams.pem, a boolean called "bFIPS_mode" must be set to "true". This flag will tell the module to run self-tests and behave according to FIPS requirements. The Crypto-Officer must also check the configuration file, ll_tunnel.conf. They must ensure that certain flags are set. The "compression zlib:9" flag will allow for maximum compression, "keylength 256" will configure the module to use keys up to 256 bits, and "keylifetime 36000" defines the lifetime of shared keys, in seconds. The Crypto-Officer must configure these flags, if they are not already set to the proper values.

In order for the module to operate as a secure tunnel, two module instances must be running: a client and a server. Client modules send data to the server module, as defined by forwarding rules, which are defined by the Crypto-Officer via the LogLogic appliance's Web UI. A forwarding rule defines a target IP address, port, and UDP/TCP mode information.

Once a forwarding rule has been defined, when initialized, the calling application, engine_stunnel, will deploy a server module, to listen for connections, and a client module, to send encrypted data over a secure tunnel to a running instance of the server module on another LogLogic appliance, defined by the forwarding rule.

If no forwarding rule is defined, the module will only start in server mode, and listen for connections.

Since the module runs automatically upon start-up, after the initial setup there will be instances of the module that are running in non-FIPS mode. The Crypto-Officer must execute the command "mtask stop" in the CLI, to unload any instances running in non-FIPS mode. Then the Crypto-Officer must execute the command "mtask start" to reinitialize the module with the FIPS-mode configuration. After this point, the module will operate in FIPS mode, automatically, until FIPS mode is disabled in the module settings.

3.2 Secure Management

This section provides guidance which ensures that the module is always operated in a secure configuration. The Crypto-Officer is responsible for understanding the module and making sure that it is running in the FIPS-Approved mode of operation.

3.2.1 Initialization

It is the Crypto-Officer's responsibility to configure the module into the FIPS-Approved mode of operation.

The module will initialize automatically, when the LogLogic appliance boots. When initializing the module, the calling application specifies which configuration file will be used for the module's configuration data. It passes information from ll_tunnel_c.conf, for client mode,or ll_tunnel.conf, for server mode, These configuration files are within the cryptographic boundary and contain basic startup configuration information for key generation and establishment.

When the module loads, it checks the settings in the dhparams.pem and compatibility.xml files. Once this is done, the module will run its Power-On Self-Tests. Upon the successful completion of self-tests, all of the FIPS-Approved cryptographic methods will be employed, and the module will be initialized in its FIPS-Approved mode of operation. At this point, in the syslog file (/var/log/sys.log), ll_tunnel will display "ll_tunnel FIPS_mode() = 1" to indicate that the module is in FIPS mode. All further actions and their success/failure messages will print to the syslog, associated with the ll_tunnel Process Identifier (PID).

If any of these files are not found, then the module will fail to properly initialize. Failure to properly initialize the module will result in the module operating in an unsupported configuration outside of the scope of its FIPS validation. In this state, all cryptographic functions will return a failure message to the syslog.

After initialization, the server module will listen for client module to establish a connection. A client module can be set to connect with a server module, at run-time, as defined by a forwarding rule that the Crypto-Officer configures. Once the connection is made, a secure tunnel is established.

3.2.2 Management

The Crypto-Officer should monitor the module's status regularly and make sure that only the services listed in Table 3 and Table 4 are being used. If any irregular activity is noticed or the module is consistently reporting errors, then LogLogic customer support should be contacted.

3.2.3 Zeroization

Most keys generated by the module are session-only keys. These are not stored by the module and only exist in volatile memory (RAM). These keys are zeroized when the software module's memory space is unloaded. This occurs when the module is restarted or shut down. The only keys that are persistent are the Diffie-Hellman Public/Private key pair on the client module hard drive and the HMAC SHA-1 integrity hash. The HMAC SHA-1 integrity hash value is stored in a file, ll_tunnel.hmacsha1, within the cryptographic boundary. The integrity hash requires uninstallation of the module to zeroize. The server-side module requires a reboot in order to zeroize the Diffie-Hellman keypair, as it resides in volatile memory. The client-side module requires a format of the host appliance's hard drive to zeroize its Diffie-Hellman keypair.

3.3 User Guidance

Only the module's tunneling services are available to the User. Users are responsible to use only the services that are listed in Table 4. Although the User does not have any ability to modify the configuration of the module, they should report to the Crypto-Officer if any irregular activity is noticed.

4 Acronyms

This section describes the acronyms.

Table 7 – Acronyms

Acronym	Definition
CFB	Cipher Feedback
CLI	Command Line Interface
CMVP	Cryptographic Module Validation Program
CSEC	Communications Security Establishment Canada
CSP	Critical Security Parameter
CPU	Central Processing Unit
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
EVA	Enterprise Virtual Appliance
FIPS	Federal Information Processing Standard
GPC	General-Purpose Computer
HMAC	(Keyed-) Hash Message Authentication Code
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
IT	Information Technology
KAT	Known Answer Test
NIST	National Institute of Standards and Technology
NVLAP	National Voluntary Laboratory Accreditation Program
OS	Operating System
PRNG	Pseudo-Random Number Generator
RAM	Random Access Memory
RNG	Random Number Generator
SHA	Secure Hash Algorithm
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VM	Virtual Machine

Prepared by:
Corsec Security, Inc.

13135 Lee Jackson Memorial Highway, Suite 220
Fairfax, VA 22033
United States of America
Phone: +1 (703) 267-6050

Email: info@corsec.com
http://www.corsec.com