Cranite™ Systems
WirelessWall Client Version 3.3

FIPS 140-2
Security Policy

Version 1.2
2/23/06

Contents

List of Figures

Revision History for Cranite Systems WirelessWall Client Version 3.3 Security Policy

Version	Date	Comments
1.0	11/22/05	Initial Release
1.1	2/15/06	Updated per CMVP comments
1.2	2/23/06	Updated per CMVP comments

1.0 Introduction

This security policy defines all security rules under which the Cranite WirelessWall Client software must operate.

The Client is a software-only *cryptographic software application* that operates on a multi-chip standalone device (PC). The logical boundary of the Client software is the self-contained compiled code that is installed by the customer or reseller into the production-quality compliant computer hardware. The physical boundary is the casing of the hardware platform on which the Client software is installed. The Client cryptographic module functions on Windows 2000/XP platforms.

The Client software and computer hardware combination operates as an *electronic encryption device*, which secures wireless network traffic between the host on which the Client is installed and the Cranite Wireless Access Controller (WAC, which is identified as the FIPS User).

Our system delivers security through five primary architectural elements:

- Defined trust relationships that ensure no single system element can compromise the integrity of the entire system.
- A role identification framework that safeguards operators' credentials regardless of the underlying system authentication mechanisms.
- Data encryption performed at Layer 2 of the Open System Interconnection model (OSI) for enhanced defense against network intrusions, denial of service attacks, and theft of data.
- Flexible security policies integrated with popular corporate directories for easy policy creation and management.
- Fine-grained packet filtering that allows authorization by server, application, protocol, or subnet.

There are three primary components that make up the WirelessWall system: the Wireless Access Controllers (FIPS User), the Clients (Client Operators are defined as FIPS Cryptographic Officers), and the Policy Server.

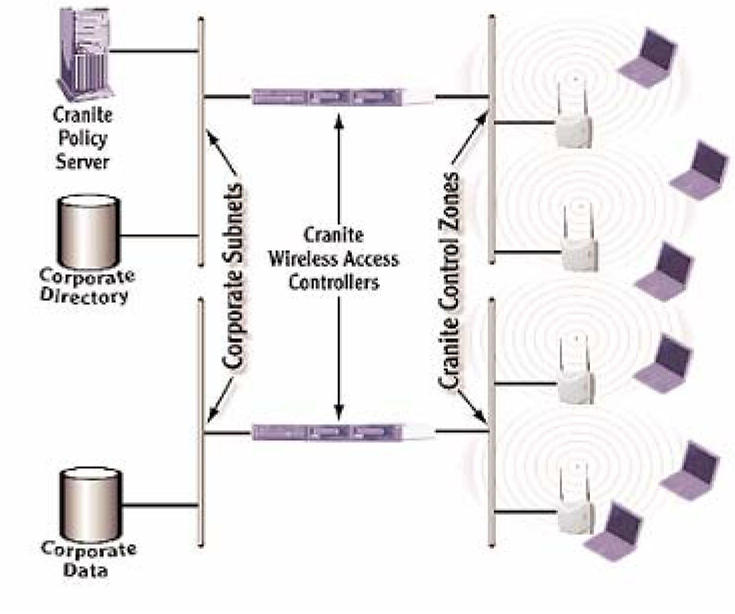

Figure 1 – Architecture Overview

Wireless Access Controllers (WAC) – Responsible for handling client identification, AES encryption/decryption, enforcing connection policies and handling transitions for roaming users. The WAC manages connections using the 802.1x standard. It also integrates with existing enterprise RADIUS servers through the Microsoft®-Challenge Handshake Authentication Protocol (MS-CHAP), Challenge Handshake Authentication Protocol (CHAP) and Password Authentication Protocol (PAP) protocols.

Client – The laptop or other device connected to the wireless network. The client contains user login and device driver code that handles user identification, state, and encryption/decryption of network traffic. The client cryptographic boundary is described later in the section titled Client Interfaces where the components of the client are described in detail.

WirelessWall Manager (also called the Policy Server) – The WirelessWall Manager is responsible for managing policies and per-user security context information to facilitate user roaming and WAC state recovery. The Manager contains a Hyper Text Markup Language (HTML) -based configuration interface to specify policies and apply policies to users and groups. The Manager integrates with existing directory services on the corporate network, including LDAP, Novell® Directory Server, Microsoft® NT Domain Server and Microsoft® Active Directory.

2.0 Client Interfaces

The WirelessWall Client is a software module that has been operationally tested on the Windows 2000 SP 4 and Windows XP SP2 platforms, but also runs on the following Operating Systems:

- Windows NT
- Pocket PC 2003
- Linux
- Mac OS X

The cryptographic boundary logically consists of the following components:

Component	Description
Cranite Service	This component performs user identification exchanges with the WirelessWall Access Controller, generates, establishes and destroys session keys and manages the configuration of the software as directed by messages it receives from the User Interface component.
Cranite Driver	This component performs bulk encryption and decryption of network traffic. Additionally, the Cranite Driver performs zeroization of keys handed to it by the Cranite Service.

Figure 2 – Component Overview (inside cryptographic logical boundary)

The module's cryptographic components communicate with the following Client-related components that reside outside of the logical cryptographic boundary:

Component	Description
Client UI	This component outside the cryptographic boundary provides the user with the ability to enable, disable or terminate the software, enter user credentials and reconfigure the network adapter selection. It informs the user of status and error states through text messages and graphic icons. This component is outside the cryptographic boundary of the Cranite client.

Gina Module	The Gina Module, also called the WirelessWall Unified Login Interface component, provides the Windows NT/2000/XP user with the option of authenticating to both the Wireless Access Controller and the Windows domain controller with a single set of user credentials. This component is outside the cryptographic boundary of the Cranite client.

Figure 3 – Components outside the cryptographic boundary

The cryptographic boundary includes all of the cryptographic services that are used by the client. The Cranite Service and the Cranite Driver are the only components that are within the Cryptographic Boundary. The Client user interface communicates with the Cranite Service through authenticated named pipes. This channel is used to push control information down to the Cranite Service and to send status information from the Cranite Service to be displayed to the operator. The Cranite Service uses secure driver binding to establish a connection with the Raw Ethernet driver. This channel is used to transport all control information during the initial dialogue with the WAC (the 'Enabling Cryptographic Software' service). This is not used after the secure session is built with the WAC. The Network Driver uses standard Windows driver binding mechanisms to establish tunnels with the Windows TCP/IP stack and with lower level drivers (usually hardware drivers). These driver bindings are kept secure and maintained by the Windows Operating System. This channel is used to send data to and from the network stack or the lower level drivers.

This diagram is accurate for all implementations and ports of the Cranite client software.

These components are implemented in the following executables:

Component	Executable				
	Windows 2000	Windows XP	Pocket PC	Linux	Mac OS X
OS version(s)					
User Interface	clientUI.exe	clientUI.exe	Wirelesswall.exe	Wirelesswall	Wirelesswall
Unified Login Interface	Csgina.dll	Csgina.dll	N/A	N/A	N/A
Daemon	clientService.exe	clientService.exe	N/A	Walld	Walld
Network Driver	Cranite.sys	Cranite.sys	Wirelesswall.dll	Walld driver	Walld driver

Figure 4 – Component Executables

The WirelessWall Client provides the following FIPS 140-2 logical interfaces:

Logical Interface	**Component Provided By**
Control Input	Control input is provided through the *User Interface* and the *Secure Data Channel with the WAC (Network Interface)* components. The User Interface controls are passed to the cryptographic boundary through a dedicated named pipe. The controls through the network interface are received via special Ethertype network packets.
Data Input	Data input consists of any incoming network traffic and occurs through the *Network Driver* component (through the Secure Data Channel with the WAC). Physically, this will be through the Ethernet device that is bound to the driver.
Data Output	Data output consists of any outgoing network traffic and occurs through the *Network driver* component. Physically, this will be through the Ethernet device that is bound to the driver.
Status Output	Status output consists of all messages output to log files by the client, or messages sent to the *User Interface* component for subsequent display on computer display

	hardware. Status is passed to the user interface components from the cryptographic boundary through a dedicated named pipe.

Figure 5 – Component Logical Interfaces

The client uses the same physical interface for data, status and control input and output. These interfaces are kept logically discrete through the use of different Ethernet frame types.

3.0 Security Level

The cryptographic module meets the overall requirements applicable to Level 1 security of FIPS 140-2. The following table shows the security levels for the different sections.

Security Requirements Section	**Level**
Cryptographic Module Specification	1
Cryptographic Module Interfaces	1
Roles and Services Identification	1
Finite State Model	1
Physical Security	N/A
Operational Environment	1
Cryptographic Key Management	1
EMI/EMC	1
Self-tests	1
Design Assurance	1
Mitigation of other attacks	N/A

Figure 6 – Security Levels

4.0 Roles and Services

The WirelessWall Client software supports two roles: Cryptographic officer and User. The Cryptographic Officer is designated as an operator on the Client possessing administrative privileges on the client's host operating system. The User role is designated as the WAC with which the Client communicates. The assumption of each role is handled via an established communication session (as in the case of the WAC) or a correct login with administrative privileges on the Client (as in the case of the Cryptographic Officer). The cryptographic officer must authenticate to the Client OS, however authentication within this product is not claimed for FIPS validation.

4.1 Cryptographic Services

The following Cryptographic Services are available on the Cranite client:

- *Enable cryptographic software* – this service enables the use of the cryptographic software. This service is run by the User role (WAC). In the FIPS deployment of the software, no traffic will pass unless this mechanism is used to first enable the cryptographic module. The client is essentially a slave to the WAC (User).
- *Show Status* – this service provides status information on the cryptographic connection. Status messages are sent outside the cryptographic boundary to the UI component.
- *Receive encrypted data* – this service processes encrypted data frames by authenticating each frame and decrypting it using the appropriate keys.

- *Send encrypted data* – this service processes unencrypted data frames destined for authenticated users by encrypting the frame using the appropriate keys and sending it to the destination Access Controller.
- *Generate a secure TLS connection* – this service generates a secure tunnel that is used to authenticate the remote Access Controller and the user and establish the session keys.
- *Generate bulk encryption keys* – this service generates the AES session keys for use during the basic operation of the cryptographic module.
- *Identify the WAC* – this service confirms the identity of the remote Access Controller through the local versification parameter file.
- *Run self-tests* – this service performs cryptographic self tests on all the cryptographic algorithms and appropriate functions within the cryptographic module. The Power-up self tests do require specific operator intervention to run, and are run each time the cryptographic software is enabled.
- *Zeroize system* - service clears and overwrites all cryptographic keys and CSPs. During this period of zeroization, the data path is inhibited to prevent data leakage.
- *Installation* – this service allows the cryptographic officer to perform installation of the Cranite WirelessWall Client.

4.2 Services Available to Each Role

Figure 7 lists the services available to each role.

CO = Cryptographic Officer (Client Operator)
U = User (WAC)

Service	Role	Control Input	Data Input	Data Output	Status Output
Enable cryptographic software	U	Message to proceed	N/A	N/A	Success or error message
Configure network Hardware	CO	Identification of hardware	N/A	N/A	Success or error message
Show status	CO	N/A	N/A	N/A	Status data as requested
Receive encrypted data	CO	N/A	Encrypted data	Decrypted data	N/A
Send encrypted data	CO	N/A	Unencrypted data	Encrypted data	N/A
Generate a secure TLS connection	CO	N/A	N/A	N/A	Success or error message
Generate bulk encryption keys	CO	Key material	N/A	N/A	Valid keys and Success or error message
Identify the WAC	CO	WAC Identification Parameter	N/A	N/A	N/A
Run Self-tests	CO	Real test or forced failure mode	N/A	N/A	(1) Test Failure or (2) Tests succeed
Zeroize system	CO	Message to release and zeroize keys	N/A	N/A	N/A
Installation, general configuration and removal of the software	CO	Input from Crypto Officer	N/A	N/A	Success or error message

Figure 7 – Services, Roles and Inputs/Outputs

4.3 Identification Methods for Each Operator Role

4.3.1 User Role

The WAC is identified to the Client cryptographic module by way of the TLS connection that is generated between the two. The Client does not function independently of the WAC.

4.3.2 Cryptographic Officer Role

The Cryptographic Officer role is available only to an operator who has been identified and authenticated to the Client operating system (outside the logical boundary).

5.0 Access Control Policy

5.1 Critical Security Parameters

The following are the critical security parameters:

- **TLS Pre-master Secret** - This key is used to generate the WAC-Client TLS Master secret.
- **WAC-Client TLS master secret** - This key is used to generate the WAC-Client TLS session keys as well as the WAC to Client AES keys.
- **WAC-Client AES keys** - These keys are used by the WAC and client to a) encrypt frames from the WAC to the client (server write key), and b) decrypt frames for messages from the client to the WAC (client write key).
- **WAC-Client AES Session HMAC keys** - These keys are used by the WAC and client to a) calculate a MAC for frames from the WAC to the client (server write key), and b) check the MAC for messages from the client to the WAC (client write key).
- **WAC-Client TLS Session TDES keys** - These keys are established as part of the TLS Protocol using cipher suite TLS_RSA_3DES_EDE_CBC mode.
- **WAC-Client TLS Session HMAC keys** - These keys are established as part of the TLS Protocol using cipher suite TLS_RSA_3DES_EDE_CBC mode.
- **Client Software HMAC-SHA1 Integrity Key -** This key is used by the module to verify the integrity of the Client software at startup.

PRNG State: This CSP is used in the generation of the WAC-Client AES and AES Session HMAC keys.

5.2 Security Parameters based on Role

CO = "Role possesses keys/CSPs on the Client (cryptographic module)"
U = "User role (WAC) establishes corresponding keys/CSPs to Client version (outside the cryptographic boundary with no direct access to these values on the cryptographic module)"

Critical Security Parameters	CO (Client)	User (WAC)
TLS Pre-master Secret	CO	
WAC-Client TLS master secret	CO	U
WAC-Client AES keys	CO	U
WAC-Client AES Session HMAC keys	CO	U
WAC-Client TLS Session TDES keys	CO	U
WAC-Client TLS Session HMAC keys	CO	U
Client Software HMAC-SHA1 Integrity Key	CO	
PRNG State	CO	

Figure 8 – Security Parameters Based on Role

5.3 Access Rights within Services

CO = Cryptographic Officer (operator of the Client)

CSPs Services	TLS Pre-master Secret	WAC-Client TLS master secret	WAC-Client AES keys	WAC-Client AES Session HMAC	WAC-Client TLS Session TDES keys	WAC-Client TLS Session HMAC keys	PRNG State
Receive encrypted data			CO	CO	CO	CO	
Send encrypted data			CO	CO	CO	CO	
Generate a secure TLS connection	CO	CO			CO	CO	
Generate bulk encryption keys		CO					CO
Zeroize system	CO	CO	CO	CO	CO		CO

Figure 9 – Services / CSPs

6.0 General Rules

The FIPS-version of the client has only one mode of operation – FIPS mode. When the cryptographic module is started (either through user direct intervention or through automatic start-up on boot), this mode of operation is entered automatically with no operator intervention. If the Client does not detect a WAC in FIPS-mode, it does not function or perform any cryptographic services.

6.1 Access Control Prior to Cryptographic Module Initialization

After the client software is installed, but before the cryptographic module is initialized, there are no cryptographic services available to operators.

6.2 Concurrent Operators

The cryptographic module does not support multiple, concurrent operators.

6.3 Software Security

The client software is written in a combination of C and C++ and operates on the following platforms: Windows 2000, Windows XP SP1, Windows XP SP2, PocketPC 2003, Macintosh OS 10.3 and 10.4, and Linux Fedora Core 1 operating systems. The software is installed in the client hardware storage medium as compiled binary executable components.

At system initialization and restart, the cryptographic module software components are each tested for integrity using HMAC to ensure the software has not been modified. If the integrity is verified, the unencrypted software is loaded into the system memory. If it fails; the cryptographic module is placed in the Crypto Failure state and can only exit that state following another system initialization or restart.

6.4 Operating System Security

The client software will operate automatically after power-up or after explicit launch by the user. The operating system is protected by the same mechanisms that are used in normal operations. This differs by platform, but standard protection mechanisms are enforced in each environment. In all cases, the host platform must be run in single user mode.

6.5 Protection of Role Identification Data

During operator role identification, passwords are masked from entry and are not echoed to the operator console.

7.0 Cryptographic Key Management

7.1 Key Generation

Once the WirelessWall Client has been enabled and a TLS handshake has begun, the following keys are generated using the FIPS-186-2 PRNG. During key generation, no traffic is allowed to pass until the Cranite Service notifies the Cranite Driver that key generation is complete. All other keys are established using TLS.

Keys	Description
WAC-Client AES keys	These are generated using the FIPS 186-2 PRNG and are used to encrypt/decrypt bulk traffic.
WAC-Client AES Session HMAC keys	These are generated using the FIPS 186-2 PRNG and used to generate and check MACs during the AES session.

Figure 10 – Key Generation

7.2 Key Storage and Zeroization

Keys are never written to permanent storage by the module. They exist only in volatile memory for the duration of their use and the relevant keys are automatically zeroized when the Client software is disabled, the TLS tunnel is taken down, the bulk encryption (AES) session expires, or the PC is rebooted. During this period of zeroization, the data path is inhibited (by setting a flag in the driver) to prevent inadvertent data leakage. The only key that exists on the hard drive is the Client Software HMAC-SHA1 Integrity Key, and it is embedded within the Client software executable binary.

7.3 Cryptographic Algorithms

The following cryptographic algorithms are used:

- FIPS-Approved Pseudo-Random Number Generator (PRNG) - Implemented FIPS 186-2 appendix 3.1 PRNG (strength at least 128 bits)
- SHA-1 - Seed key byte size: 20
- HMAC-SHA1 - Key Size == Block Size == 64 bytes, MAC size: 20 bytes
- RSA –
    - Note: RSA 2048 bit modulus digital signature generation and verification is latent functionality within this module, and despite possessing an algorithm certificate, this form of RSA is not currently utilized.
- AES - ECB,CTR modes
- TDES – TCBC mode

Non-FIPS Approved Algorithms used:

- RSA public key cipher (used within the TLS key agreement protocol; key wrapping; key establishment methodology provides 128 bits of encryption strength) – may be used in FIPS-mode
- MD5 (used in TLS key establishment) – may be used in FIPS-mode

7.4 Cryptographic Self-Tests

Self-tests are initiated when the cryptographic module is loaded. Each module within the software cryptographic boundary that contains cryptographic algorithms initiates the cryptographic self-test at load time. The following table describes the self tests performed when the modules are loaded:

Component	**Cryptographic Algorithm**	**CypherSuite/Mode**	**Test Type**
Cranite Service	TDES	TCBC	Known Answer Test
Cranite Service	RSA	Encryption	Known Answer Test
Cranite Service	PRNG	FIPS 186-2 appendix 3.1 PRNG	Known Answer Test
Cranite Driver	AES	ECB, CTR 128 bit	Known Answer Test
Cranite Driver	HMAC SHA-1	Size == Block Size == 64 bytes, MAC size: 20 bytes	Known Answer Test
Cranite Service	PRNG Test	Validate size; check that old 160 bit value is not equal to the new 160 bit value.	Conditional Self-Test

Figure 11 – Self-Tests

These tests are performed immediately on application startup and prior to the encryption or decryption of any network traffic. The tests are initiated automatically without any user initiation. In addition, the client performs a continuous PRNG test to ensure that each call to the PRNG yields a different result from the previous call. If any of the tests fail, the client displays an error message and terminates. The error results are written to a log file named 'client-debug.txt' in the install directory of the application. If any self test fails, an error message will be displayed in the system event manager and the system will disable all cryptographic functions by unloading the cryptographic module and placing the Cryptographic Module in the Crypto Failure State. In this state, data traffic is inhibited.

To initiate self tests on-demand, the Cryptographic Officer should perform the following steps:

1. Set the WirelessWall Client into the "WirelessWall Off" state.
2. Stop the client application through normal Operating systems mechanisms.
3. Stop the CraniteService using the appropriate action for the operating system.
4. Start the CraniteService using the appropriate action for the operating system.
5. Start the client application through normal Operating systems mechanisms.
6. Set the WirelessWall Client into the "WirelessWall On" state.

7.5 Continuous PRNG Test

The FIPS-approved PRNG (FIPS 186-2 appendix 3.1 PRNG) used in the Cryptographic Module executes a continuous PRNG test which ensures that each call to the random number generator yields a different result from the previously generated 160-bit block. If the same 160-bit block is returned, the Cryptographic Module is placed in the Crypto Failure State, the cryptographic functions are unloaded and all data traffic is inhibited.

7.6 Determining the Status of the Cryptographic Module

In order to determine the status of the cryptographic module, the cryptographic officer must first authenticate him/herself to the client and then verify that the Cranite Service is currently operating. On Windows operating systems, the cryptographic officer should launch the "services" tool. On Windows XP, this tool is found in the Control Panel -> Administrative Tools path. The service called "CraniteService" should be stopped (no status will be displayed) if the tests have failed and "Started" if the tests have passed.

The cryptographic officer can also view the status of the cryptographic module by reviewing the log files stored in the client install directory or by examining the event manager.

7.7 Error State Handling

Should any cryptographic processing encounter an error condition that places the cryptographic module in the Cryptographic Failure State, the error condition will be written to the log files and the cryptographic module will be disabled by being unloaded from system memory when the service is terminated. In this state, all data traffic is inhibited. The only way to recover from the Error State is by re-initiating the system, as specified above.

8.0 Physical Security Policy

The WirelessWall Client software is installed by the customer or VAR on production-quality, FCC-certified hardware (such as a PC), which also defines the module's physical boundary. The minimum requirement for the computer hardware is as follows:

- X86 processor system board
- CD-ROM
- 128 MB RAM

- One 802.11 Network Interface Card
- 3 Gb Hard drive
- Opaque Enclosure
- Overall system must be FCC Certified to Part 15, Subpart B, Class B.

These hardware requirements ensure that any hardware platform on which the WirelessWall Client is installed complies with the requirement for FIPS Security Level 1. The physical security of a deployed machine is determined by the customer's security policy. The module relies on the standard PC computer case to offer physical protection. All components (circuit boards, components, casing) of the PC are standard production-grade quality.

On Windows, tests were performed on Microsoft Windows XP SP2 with the latest patches, and Windows 2000 Service Pack 4 (SP4).

9.0 Mitigation of Other Attacks

The cryptographic module is not designed to mitigate attacks outside the scope of FIPS 140-2.

10.0 Glossary

AES - Advanced Encryption Standard
CA - Certificate Authority
CBC - Cipher Block Change
CHAP - Challenge Handshake Authentication Protocol
CSP - Cryptographic Security Parameter
CTR - Counter Mode
ECB - Electronic Code Block
EDE - Encryption Decryption Encryption
FCC - Federal Communications Commission
FIPS - Federal Information Processing Standards
HMAC - Keyed-Hashing for Message Authentication
IP - Internet Protocol
IPC - Inter-Process Communication
LDAP - Lightweight Directory Access Protocol
MAC - Medium Access Control
MD5 - Message-Digest Algorithm
MIC - Message Integrity Check
MS-CHAP - Microsoft Challenge Handshake Authentication Protocol
OS - Operating System
OSI - Open Systems Interconnection
PAP - Password Authentication Protocol
PC - Personal Computer
PRNG - Pseudo Random Number Generator
RADIUS - Remote Authentication Dial-In User Service
RSA - Rivest-Shamir-Adelman
SHA-1 - Secure Hash Algorithm
TCBC TDEA - Cipher Block Chaining Mode of Operation
TDEA - Triple Data Encryption Algorithm
TDES - Triple Data Encryption Standard
TLS - Transport Layer Security
VAR - Value Added Reseller

WAC - Wireless Access Controller

11.0 Document Reference List

Derived Test Requirements for FIPS Pub 140-2, Security Requirements for Cryptographic Modules NIST, March 2004
Approved Random Number Generators for FIPS Pub 140-2, Security Requirements for Cryptographic Modules, NIST, January 2005
Approved Key Establishment Techniques for FIPS Pub 140-2, Security Requirements for Cryptographic Modules, NIST, June 2005
Key Management Guideline, Second Draft NIST, June 3, 2002
RFC2246, The TLS Protocol, Version 1.0
RFC2716, PPP EAP TLS Authentication Protocol
***Building Secure Software* (by John Viega and Gary McGraw, Addison-Wesley, 2002)**
***SSL and TLS Designing and Building Secure Systems* (by Eric Rescorla, Addison-Wesley, 1972)**
FIPS Pub 186-2, Digital Signature Standard (DSS), NIST, January 27, 2000
***Client Design Document*, Cranite Systems, updated August 2005**
***Key Management –Client Document*, Cranite Systems, updated August 2005**