Boeing
Secure Network Server
(SNS-3010, SNS-3110, and SNS-3210)
Security Target

Version 2.5
2/3/2011

Prepared for and Revised By:
The Boeing Company
P.O. Box 3999, M/S 88-12
Seattle, WA 98124-2499

LIST OF FIGURES

LIST OF TABLES

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is the Boeing Secure Network Server (SNS) provided by The Boeing Company. The TOE is a network appliance, more specifically a guard that serves to control the flow of information between attached subscriber devices. It is capable of controlling information flows based on information in packet headers, packet contents, and security labels associated with packets and the subscribers. Each subscriber is configured with a sensitivity label range that limits (via Mandatory Access Controls (MAC)) the labels that can be associated with information that can come from or go to a given subscriber. In addition to MAC, the SNS can be configured to limit the flow of information based on packet attributes (e.g., addresses), contents, and other datagram characteristics as well as to constrain the flow of information to mitigate the potential for covert channels. The information flow policies are managed by defined administrators that can manage subscriber devices and the policy rules to affect an information flow policy suitable for their specific application.

The Security Target contains the following additional sections:

- TOE Description (Section 2)
- Security Environment (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
- Protection Profile Claims (Section 7)
- Rationale (Section 8).

1.1 Security Target, TOE and CC Identification

ST Title – Boeing Secure Network Server (SNS-3010, SNS-3110, and SNS-3210) Security Target

Firmware Version: 3.10.7

ST Version – Version 2.3

ST Date – 1/20/11

TOE Identification – Boeing Secure Network Server (SNS-3010, SNS-3110, and SNS-3210)

TOE Developer – The Boeing Company

Evaluation Sponsor – The Boeing Company

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 2.3, August 2005

1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Requirements, Version 2.3, August 2005.
    - Part 2 Conformant
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Requirements, Version 2.3, August 2005.

- Part 3 Conformant
- Assurance Level: EAL 5 augmented with ACM_AUT.2, ACM_CAP.5, ADO_DEL.3, ADV_HLD.4, ADV_IMP.3, ADV_INT.3, ADV_LLD.2, ADV_RCR.3, ALC_DVS.2, ALC_FLR.2, ALC_LCD.3, ALC_TAT.3, ATE_COV.3, ATE_DPT.3, ATE_FUN.2, AVA_CCA.2, and AVA_MSU.3
- Strength of Function Claim: SOF-high

1.3 Conventions, Terminology, Acronyms

This section specifies the formatting information used in the Security Target.

1.3.1 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
    - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a letter placed at the end of the component. For example FDP_ACC.1a and FDP_ACC.1b indicate that the ST includes two iterations of the FDP_ACC.1 requirement, a and b.
    - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]).
    - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
    - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.3.2 Terminology and Acronyms

ARP	Address Resolution Protocol
BIOS	basic input output system
BIT	built in test
CD	carrier detect
CIPSO	Common IP Security Option
CRC	cyclic redundancy check
DAC	discretionary access control
DoD	Department of Defense
DOI	domain of interpretation
EPROM	erasable, programmable, read-only memory
GDT	global descriptor table
HTTP	Hyper-Text Transfer Protocol
HTTPS	HTTP Secure

ICMP	Internet Control Message Protocol
IDT	interrupt descriptor table
IOSYS	input/output system
IP	Internet Protocol
ITC	intertask communication
LAN	local area network
Labeled Interface	A physical port on the TOE to which the attached subscriber device sends and receives datagrams with sensitivity labels.
LDT	local descriptor table
MAC	Mandatory Access Control
Mbps	megabits per second
MLS	multilevel secure
NA	network administrator
NI	network interface
NM	network management
NMI	NM interface
NTCB	network trusted computing base
PIC	Programmable Interrupt Controller
RARP	Reverse Address Resolution Protocol
RDP	reliable datagram protocol
SA	security administrator
SSA	Super-SA
SNMP	Simple Network Management Protocol
SNS	Secure Network Server
SM	SNS management
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TNI	Trusted Network Interpretations
TSS	task state segment
UDP	User Datagram Protocol
Unlabeled Interface	A physical port on the TOE to which the attached subscriber device sends and receives datagrams without sensitivity labels.

2. TOE Description

The Target of Evaluation (TOE) is Boeing Secure Network Server (SNS), versions 3010, 3110, and 3210. Each version of the TOE utilizes the same software and bios; the primary differences being physical. The 3010 version includes a 4U rack-mountable chassis and the 3210 version includes a 2U rack-mountable chassis. The 3110 version includes a 2U flight-worthy chassis that, unlike the other two versions, utilizes a solid-state hard drive.

2.1 TOE Overview

The TOE is a guard primarily designed to control the flow of information among attached subscriber devices. Each subscriber is configured with a sensitivity label range that limits (via Mandatory Access Controls (MAC)) the labels that can be associated with information that can come from or go to a given subscriber. In addition to MAC, the TOE can be configured to limit the flow of information based on packet attributes (e.g., addresses), contents, and other datagram characteristics as well as to constrain the flow of information to mitigate the potential for covert channels. The information flow policies are managed by defined administrators that can manage subscriber devices and the policy rules to affect an information flow policy suitable for their specific application.

2.2 TOE Architecture

The Boeing SNS is a network appliance running on a custom kernel that runs on COTS hardware (with a custom BIOS) based on the Intel Pentium 4 processor. The SNS utilizes the Intel Pentium 4 ring architecture to separate its own functions resulting in a well-layered design that implements a least privilege principle. Each appliance supports serial devices (consoles) and network devices (subscriber devices).

The TOE consists of hardware and firmware, composing one or more Boeing SNS appliances with one acting as a Network Management (NM) appliance. The distributed TOE components are always synchronized with the NM and are managed from the central NM appliance. Also, the connections among the distributed TOE components must be distinct from the connections to the subscriber devices since the entire connection media must be protected to protect sensitive TOE communications. The TOE boundary is everything inside the TSF as shown in Figure 2.

2.2.1 Physical Boundaries

Physically, there may be three consoles (connected via serial ports): utility, SA, and NA. Alternately, a single console (or attached keyboard and monitor) can be configured to use defined keyboard key sequences (Ctrl-T, Ctrl-Y, Ctrl-N) to logically switch between three consoles. The other important interfaces are a dedicated Ethernet port for SNS-to-SNS communication and additional Ethernet ports to the subscriber devices outside the TOE. The consoles offer management functions and the subscriber interfaces internal to the TOE offer controlled information flow among the attached subscriber devices outside the TOE. Figure 1 shows a sample SNS configuration. Figure 2 shows the major architectural components and the TOE boundary.

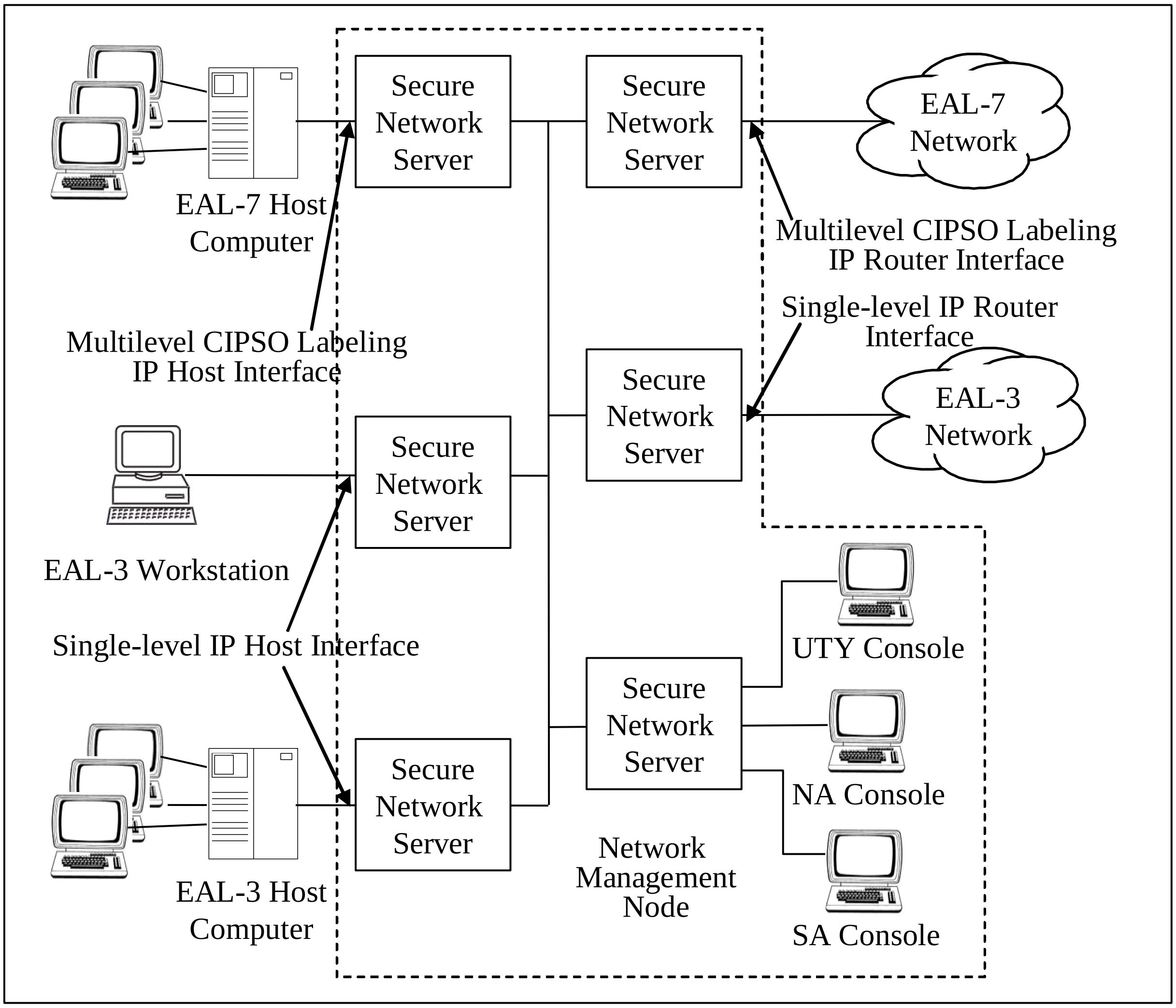

Figure 1 Sample SNS Configuration

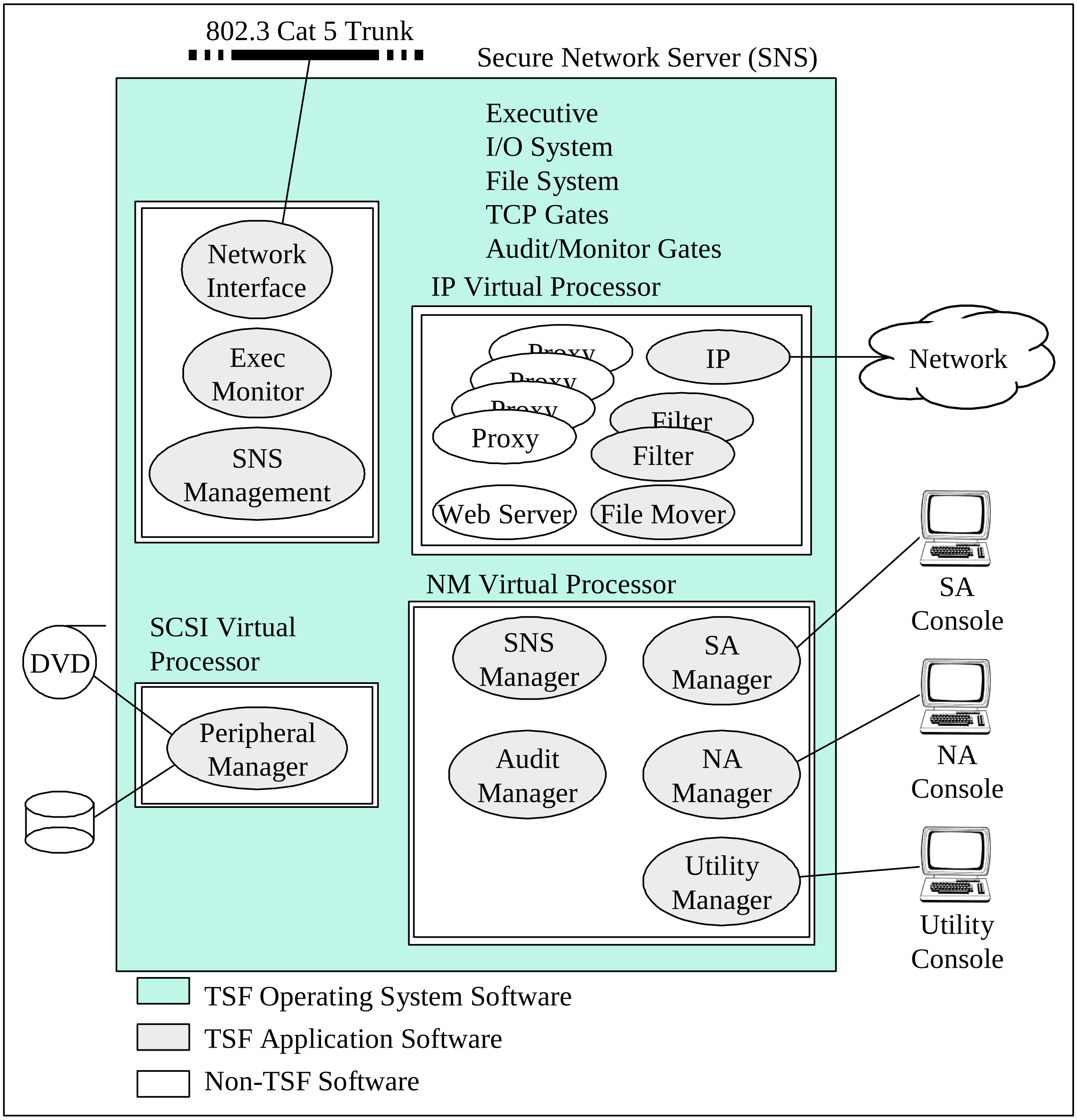

Figure 2 System Components

2.2.2 Logical Boundaries

This section identifies the security functions that SNS provides.

2.2.2.1 Security audit

The Boeing SNS generates audit events for security relevant events, including covert channel indicators. The audit events are stored and protected, and forwarded to the NM for review and archival purposes. The SNS sends warning when the audit storage capacity is nearing or has exceeded its capacity and it can be configured to automatically overwrite events or to stop operations altogether until the situation is remedied.

2.2.2.2 User data protection

The Boeing SNS is design primarily to control the flow of information between subscriber devices. It enforces a rich set of information flow policies including mandatory access controls based on subscriber sensitivity labels, packet filtering, and content filtering (binary messages). It also provides routing and processing functionality to offer static routing, multicast support, and ICMP.

2.2.2.3 Identification and authentication

While all users (administrators) and subscriber devices are identified by the SNS, it also requires that administrators are authenticated at an appropriate management console prior to offering management functions. This is accomplished by managing user definitions, including user identities, roles, and associated authentication data (i.e., passwords).

In order to help mitigate attempts to bypass the authentication mechanisms, the Boeing SNS informs users each time they log in of the last time they successfully logged in, the number of unsuccessful logins that have occurred since the last successful login, and the time of the last unsuccessful login attempt.

2.2.2.4 Security management

The Boeing SNS offers command line interfaces for the management of the TOE Security Functions. There are three defined roles: Network Administrator (NA), Security Administrator (SA), and Super-SA (SSA). The Super-SA primarily manages the administrator accounts, the SA primarily manages the security functions, and the NA primarily manages the general operational capabilities of the TOE. The Super-SA can do everything the SA can do, and additionally is able to create administrator accounts. Each administrator must log into the appropriate console before applicable functions can be accessed.

2.2.2.5 Protection of the TSF

The Boeing SNS is designed around a custom operating kernel that makes use of the ring architecture offered by Intel Pentium 4 processors to protect itself and to separate itself to implement a least privilege principle. All traffic flowing through the TOE is subject to its security policies. Furthermore, the TOE includes self tests that run at initial start-up and also periodically when the TOE is operational. The TOE also includes failure detection and recovery features to ensure that it continues to operate correctly when recoverable failures occur and to ensure that it shuts down when necessary when manual recovery becomes necessary.

The Boeing SNS is designed so that a given part of a distributed SNS system can continue to operate properly when some other system components (i.e., other SNSs) fail. It is also designed to limit the throughput of a given device to protect itself and other network components as may be necessary.

2.3 TOE Documentation

Boeing offers administrator guidance and has subjected other Boeing SNS-related documentation for the purpose of evaluation. See section 6.2 for more details.

3. Security Environment

This section defines threats countered and assumptions made about the environment of the TOE.

Note that the claimed assurance level (EAL 5 augmented with ACM_AUT.2, ACM_CAP.5, ADO_DEL.3, ADV_HLD.4, ADV_IMP.3, ADV_INT.3, ADV_LLD.2, ADV_RCR.3, ALC_DVS.2, ALC_FLR.2, ALC_LCD.3, ALC_TAT.3, ATE_COV.3, ATE_DPT.3, ATE_FUN.2, AVA_CCA.2, and AVA_MSU.3) also serves to define the environment since it affects the strength of the security functional claims.

3.1 Threats

T.AUDIT	Attempts to violate TOE security policies may go undetected or users may not be accountable for security-relevant actions they perform.
T.FILTER	Inappropriate network traffic may enter or leave a protected network.
T.I&A	Unauthorized users may be able to inappropriately configure the TOE or access sensitive TOE data.
T.MAC	Classified information may be inappropriately accessed by entities that do not have appropriate clearances.
T.OPERATE	The TOE may fail to provide or enforce its security functions due to failure or malicious attacks against its security mechanisms.

3.2 Assumptions

A.ADMIN	The TOE administrators are competent, adhere to the applicable guidance, and are not willfully negligent or malicious.
A.COMMS	The TOE is able to communicate with its attached subscriber devices.
A.FLOW	Protected information does not flow among the network subscribers unless it passes through the TOE.
A.PHYSEC	The TOE is physically secure; specifically it, including the communication media among distributed parts of the TOE, is protected from physical tampering of itself or its physical connections to its environment (subscriber devices).
A.SUBSCRIBE	A process outside the scope or control of the TOE is used to determine the attributes (e.g., sensitivity ranges) of attached subscriber devices.

4. Security Objectives

This section defines the security objectives for the TOE and its environment that is necessary in order to address the environment as characterized in the previous section.

4.1 Security Objectives for the TOE

O.AUDLOS	The TSF shall be configurable to limit the potential loss of audit information.
O.AUDREC	The TOE shall provide a means to record an audit trail of security-related events, with accurate dates and times.
O.AUDREV	The TSF shall protect the audit trail so that only an authorized administrator can access the audit trail.
O.AUDTHR	The TSF shall allow audit thresholds to be defined that will trigger alarms when attempted policy violations exceed the defined thresholds.
O.FILTER1	The TOE shall allow (only) an authorized administrator to explicitly define information filtering rules.
O.FILTER2	The TOE shall restrict the flow of information among subscriber devices based on filtering rules based on information headers and content established by the authorized administrator.
O.IDAUTH	The TOE shall uniquely identify and authenticate the claimed identity of all administrators before granting access to TOE functions related to the assumed administrator role.
O.IMPEXP	The TOE shall import and export labeled and unlabelled data according to the sensitivity labels associated with attached subscriber devices.
O.MAC1	The TOE shall allow (only) an authorized administrator to assign sensitivity labels to subscriber devices.
O.MAC2	The TOE shall restrict the flow of information between attached subscriber devices so that information from one subscriber can be sent to another subscriber only if the sensitivity level of the information is within the range of sensitivity labels the receiving subscriber device is allowed to process.
O.PROTECT	The TOE shall ensure that its functions are always invoked and that it is resistant to potential attacks against its security functions.
O.RECOVER	The TOE shall remain secure and be able to recover from failure conditions and will continue to operate when possible.
O.SELFTEST	The TOE shall test its own operation in order to detect potential failures.

4.2 Security Objectives for the Environment

OE.ADMIN The TOE administrators will be competent, adhere to the applicable guidance, and will not be willfully negligent or malicious.

OE.COMMS The TOE will be able to communicate with its attached subscriber devices.

OE.FLOW Protected information does not flow among the network subscribers unless it passes through the TOE.

OE.PHYSEC The TOE, and the communication media among distributed parts of the TOE, will be physically protected from physical tampering of itself or its physical connections to its environment.

OE.SUBSCRIBE A process outside the scope or control of the TOE will be used to determine the attributes (e.g., sensitivity ranges) of attached subscriber devices.

5. IT Security Requirements

This section defines the security functional and assurance requirements satisfied by Boeing Secure Network Server (SNS). Each of these requirements has been drawn from version 2.3 of the Common Criteria, Parts 2 and 3 and has been completed in this Security Target as necessary.

5.1 TOE Security Functional Requirements

The following table describes the SFRs that are candidates to be satisfied by Boeing SNS.

Requirement Class	Requirement Component
FAU: Security audit	FAU_ARP.1: Security alarms
	FAU_GEN.1: Audit data generation
	FAU_SAA.1: Potential violation analysis
	FAU_SAR.1: Audit review
	FAU_SAR.2: Restricted audit review
	FAU_SEL.1: Selective audit
	FAU_STG.2: Guarantees of audit data availability
	FAU_STG.3: Action in case of possible audit data loss
	FAU_STG.4: Prevention of audit data loss
FDP: User data protection	FDP_ETC.1: Export of user data without security attributes
	FDP_ETC.2: Export of user data with security attributes
	FDP_IFC.2: Complete information flow control
	FDP_IFF.2: Hierarchical security attributes
	FDP_IFF.4: Partial elimination of illicit information flows
	FDP_ITC.1: Import of user data without security attributes
	FDP_ITC.2: Import of user data with security attributes
	FDP_RIP.2: Full residual information protection
FIA: Identification and authentication	FIA_AFL.1: Authentication failure handling
	FIA_ATD.1: User attribute definition
	FIA_SOS.1: Verification of secrets
	FIA_UAU.1: Timing of authentication
	FIA_UAU.7: Protected authentication feedback
	FIA_UID.2: User identification before any action
FMT: Security management	FMT_MOF.1: Management of security functions behaviour
	FMT_MSA.1: Management of security attributes
	FMT_MSA.3: Static attribute initialization
	FMT_MTD.1: Management of TSF data
	FMT_SAE.1: Time-limited authorization
	FMT_SMF.1: Specification of Management Functions
	FMT_SMR.2: Restrictions on security roles
	FMT_SMR.3: Assuming roles
FPT: Protection of the TSF	FPT_AMT.1: Abstract machine testing
	FPT_FLS.1: Failure with preservation of secure state
	FPT_RCV.3: Automated recovery without undue loss
	FPT_RCV.4: Function recovery
	FPT_RVM.1: Non-bypassability of the TSP
	FPT_SEP.3: Complete reference monitor
	FPT_STM.1: Reliable time stamps
	FPT_TDC.1: Inter-TSF basic TSF data consistency
	FPT_TST.1: TSF testing

FRU: Resource utilization	FRU_FLT.2: Limited fault tolerance
	FRU_RSA.1: Maximum quotas
FTA: TOE access	FTA_TAH.1: TOE access history

Table 1 TOE Security Functional Components

5.1.1 Security audit (FAU)

5.1.1.1 Security alarms (FAU_ARP.1)

FAU_ARP.1.1 The TSF shall take [**display a warning on the SA status console, deny the audited action, and for covert storage channel related audit events block further actions until the SA/SSA-configured time duration has expired**] upon detection of a potential security violation.

5.1.1.2 Audit data generation (FAU_GEN.1)

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events: a) Start-up and shutdown of the audit functions; b) All auditable events for the [***minimum***] level of audit; and c) [**use of covert channel related mechanisms**].

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information: a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [**no additional information**].

5.1.1.3 Potential violation analysis (FAU_SAA.1)

FAU_SAA.1.1 The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the TSP.

FAU_SAA.1.2 The TSF shall enforce the following rules for monitoring audited events: a) Accumulation or combination of [**audit events**] known to indicate a potential security violation; b) [**accumulation of covert channel related audit events within an SA-configured period**].

5.1.1.4 Audit review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide [**the SA and Super-SA**] with the capability to read [**all audit data**] from the audit records.

FAU_SAR.1.2 The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

5.1.1.5 Restricted audit review (FAU_SAR.2)

FAU_SAR.2.1 The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

5.1.1.6 Selective audit (FAU_SEL.1)

FAU_SEL.1.1 The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attributes: a) [***event type***] b) [**sensitivity level**].

5.1.1.7 Guarantees of audit data availability (FAU_STG.2)

FAU_STG.2.1 The TSF shall protect the stored audit records from unauthorised deletion.

FAU_STG.2.2 The TSF shall be able to [***prevent***] unauthorised modifications to the audit records in the audit trail.

FAU_STG.2.3 The TSF shall ensure that [**all**] audit records will be maintained when the following conditions occur: [***audit storage exhaustion***].

5.1.1.8 Action in case of possible audit data loss (FAU_STG.3)

FAU_STG.3.1 The TSF shall take [**action to warn the SA console admin**] if the audit trail exceeds [**90% of its capacity**].

5.1.1.9 Prevention of audit data loss (FAU_STG.4)

FAU_STG.4.1 The TSF shall [***'prevent auditable events, except those taken by the authorised user with special rights'***] **at the discretion of the SA, or SSA,** and [**take no additional action**] if the audit trail is full.

5.1.2 User data protection (FDP)

5.1.2.1 Export of user data without security attributes (FDP_ETC.1)

FDP_ETC.1.1 The TSF shall enforce the [**Single-level Subscriber Information Flow Policy**] when exporting user data, controlled under the SFP(s), outside of the TSC.

FDP_ETC.1.2 The TSF shall export the user data without the user data's associated security attributes.

5.1.2.2 Export of user data with security attributes (FDP_ETC.2)

FDP_ETC.2.1 The TSF shall enforce the [**Multi-level Subscriber Information Flow Policy**] when exporting user data, controlled under the SFP(s), outside of the TSC.

FDP_ETC.2.2 The TSF shall export the user data with the user data's associated security attributes.

FDP_ETC.2.3 The TSF shall ensure that the security attributes, when exported outside the TSC, are unambiguously associated with the exported user data.

FDP_ETC.2.4 The TSF shall enforce the following rules when user data is exported from the TSC: [**the label on the data must be within the range of sensitivity levels assigned to the associated subscriber device**].

5.1.2.3 Complete information flow control (FDP_IFC.2)

FDP_IFC.2.1 The TSF shall enforce the [**Single-level Subscriber Information Flow Policy and Multi-level Subscriber Information Flow Policy**] on [**users (subscriber devices) and data (datagrams)**] and all operations that cause that information to flow to and from subjects covered by the SFP.

FDP_IFC.2.2 The TSF shall ensure that all operations that cause any information in the TSC to flow to and from any subject in the TSC are covered by an information flow control SFP.

5.1.2.4 Hierarchical security attributes (FDP_IFF.2)

FDP_IFF.2.1 The TSF shall enforce the [one of the following policies: **Single-level Subscriber Information Flow Policy or Multi-level Subscriber Information Flow Policy, dependent upon interface setting**] based on the following types of subject and information security attributes: [**subjects: subscriber devices with identities and defined sensitivity label range and information: datagrams with protocol number, source address, destination address, TCP/UDP source and destination ports, ICMP type, sensitivity label, and payload**].

FDP_IFF.2.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules, based on the ordering relationships between security attributes hold: [

> **a) datagrams will be accepted from a given subscriber device only if its sensitivity label is within the range of sensitivity labels assigned to that device; and**
> **b) datagrams will be sent to a given subscriber device only if its sensitivity label is within the range of sensitivity labels assigned to that device; and**
> **c) datagrams will be accepted only if they are allowed according to the packet filtering rules based on protocol, source and destination addresses, source and desitnation ports, and ICMP type**].

FDP_IFF.2.3 The TSF shall enforce the [**no additional rules**].

FDP_IFF.2.4 The TSF shall provide the following [**payload-based filters and transformations will be enforced when configured for a given port:**

a) For all messages,

1) the message will be accepted only if it satisfies a defined structure, any fields are allowed, any fields have allowed values, no dirty words are present in message, and all text fields are determined to be unlikely to contain encoded text;

2) the message will be transformed such that specified message fields will be zeroed out, or for text fields, replaced with spaces (0x20)].

FDP_IFF.2.5 The TSF shall explicitly authorise an information flow based on the following rules: [**none**].

FDP_IFF.2.6 The TSF shall explicitly deny an information flow based on the following rules: [**none**].

FDP_IFF.2.7 The TSF shall enforce the following relationships for any two valid information flow control security attributes:

a) There exists an ordering function that, given two valid security attributes, determines if the security attributes are equal, if one security attribute is greater than the other, or if the security attributes are incomparable; and

b) There exists a 'least upper bound' in the set of security attributes, such that, given any two valid security attributes, there is a valid security attribute that is greater than or equal to the two valid security attributes; and

c) There exists a 'greatest lower bound' in the set of security attributes, such that, given any two valid security attributes, there is a valid security attribute that is not greater than the two valid security attributes.

5.1.2.5 Partial elimination of illicit information flows (FDP_IFF.4)

FDP_IFF.4.1 The TSF shall enforce the [**Single-level Subscriber Information Flow Policy and Multi-level Subscriber Information Flow Policy**] to limit the capacity of [**covert storage and timing channels**] to a [**SA/SSA-controllable maximum capacity**].

FDP_IFF.4.2 The TSF shall prevent [**all covert storage channels in excess of the SA/SSA-defined capacity**].

5.1.2.6 Import of user data without security attributes (FDP_ITC.1)

FDP_ITC.1.1 The TSF shall enforce the [**Single-level Subscriber Information Flow Policy**] when importing user data, controlled under the SFP, from outside of the TSC.

FDP_ITC.1.2 The TSF shall ignore any security attributes associated with the user data when imported from outside the TSC.[**Security attributes that are valid for the device are preserved on input**.]

FDP_ITC.1.3 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TSC: [**the data will be labeled with the sensitivity label assigned to its associated subscriber device**].

5.1.2.7 Import of user data with security attributes (FDP_ITC.2)

FDP_ITC.2.1 The TSF shall enforce the [**Multi-level Subscriber Information Flow Policy**] when importing user data, controlled under the SFP, from outside of the TSC.

FDP_ITC.2.2 The TSF shall use the security attributes associated with the imported user data.

FDP_ITC.2.3 The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

FDP_ITC.2.4 The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.

FDP_ITC.2.5 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TSC: [**the label on the data must be within the range of sensitivity levels assigned to the associated subscriber device**].

5.1.2.8 Full residual information protection (FDP_RIP.2)

FDP_RIP.2.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [***allocation of the resource to***] all objects.

5.1.3 Identification and authentication (FIA)

5.1.3.1 Authentication failure handling (FIA_AFL.1)

FIA_AFL.1.1 The TSF shall detect when [***1***] unsuccessful authentication attempts occur related to [**user authentication**].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [**impose a delay until authentication can be attempted again, beginning with a SA/SSA-configured delay and doubling it with each successive failed authentication attempt**].

5.1.3.2 User attribute definition (FIA_ATD.1)

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: [**identity, roles, and authentication data**].

5.1.3.3 Verification of secrets (FIA_SOS.1)

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet [**a SA/SSA-configured minimum password length and password composition of 1, 2, 3, or 4 of the following classes: lower case alphabetic, upper case alphabetic, numeric, and non-alpha-numeric characters**].

5.1.3.4 Timing of authentication (FIA_UAU.1)

FIA_UAU.1.1 The TSF shall allow [**a) datagrams to be accepted from a given subscriber device if its sensitivity label is within range of sensitivity labels assigned to that device; b) datagrams to be sent to a given subscriber device if its sensitivity label is within the range of sensitivity labels assigned to that device; c) IP type and addresses to be validated against the IP DAC rule assigned to the subscriber**] on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

5.1.3.5 Protected authentication feedback (FIA_UAU.7)

FIA_UAU.7.1 The TSF shall provide only [**obscured feedback**] to the user while the authentication is in progress.

5.1.3.6 User identification before any action (FIA_UID.2)

FIA_UID.2.1 The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

5.1.4 Security management (FMT)

5.1.4.1 Management of security functions behaviour (FMT_MOF.1)

FMT_MOF.1.1 The TSF shall restrict the ability to [***modify the behavior of***] the functions [**Identification and Authentication, Audit, Single-level Subscriber Information Flow Policy, and Multi-level Subscriber Information Flow Policy**] to [**Super-SA, SA, and NA**].

5.1.4.2 Management of security attributes (FMT_MSA.1)

FMT_MSA.1.1 The TSF shall enforce the [**Single-level Subscriber Information Flow Policy and Multi-level Subscriber Information Flow Policy**] to restrict the ability to [***modify***] the security attributes [**sensitivity labels**] to [**the SA and SSA**].

5.1.4.3 Static attribute initialization (FMT_MSA.3)

FMT_MSA.3.1 The TSF shall enforce the [**Single-level Subscriber Information Flow Policy and Multi-level Subscriber Information Flow Policy**] to provide [***restrictive***] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [**NA, SA and SSA**] to specify alternative initial values to override the default values when an object or information is created.

5.1.4.4 Management of TSF data (FMT_MTD.1)

FMT_MTD.1.1 The TSF shall restrict the ability to [***modify, delete and [create]***] the [SA and NA **user definitions**] to [**the Super-SA**].

5.1.4.5 Time-limited authorization (FMT_SAE.1)

FMT_SAE.1.1 The TSF shall restrict the capability to specify an expiration time for [**password and user accounts**] to [**the Super-SA**].

FMT_SAE.1.2 For each of these security attributes, the TSF shall be able to [**deny subsequent user authentication]**after the expiration time for the indicated security attribute has passed.

5.1.4.6 Specification of Management Functions (FMT_SMF.1)

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: [**management of the Single-level Subscriber Information Flow Policy and Multi-level Subscriber Information Flow Policy including associated attributes, audit management and review, user attribute management, Identification and Authentication configuration**].

5.1.4.7 Restrictions on security roles (FMT_SMR.2)

FMT_SMR.2.1 The TSF shall maintain the roles: [**Network Administrator (NA), Security Administrator (SA) and Super-SA**].

FMT_SMR.2.2 The TSF shall be able to associate users with roles.

FMT_SMR.2.3 The TSF shall ensure that the conditions [**the SA and Super-SA can log into only the utility and SA consoles and the NA can log into only the NA console**] are satisfied.

5.1.4.8 Assuming roles (FMT_SMR.3)

FMT_SMR.3.1 The TSF shall require an explicit request to assume the following roles: [**NA, SA, and Super-SA**].

5.1.5 Protection of the TSF (FPT)

5.1.5.1 Abstract machine testing (FPT_AMT.1)

FPT_AMT.1.1 The TSF shall run a suite of tests [***during initial start-up* and *periodically during normal operation***] to demonstrate the correct operation of the security assumptions provide by the abstract machine that underlies the TSF.

5.1.5.2 Failure with preservation of secure state (FPT_FLS.1)

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: [**power failures, disk read and write failures, and memory read and write failures**].

5.1.5.3 Automated recovery without undue loss (FPT_RCV.3)

FPT_RCV.3.1 When automated recovery from [**loss of power or reset during database operations**] is not possible, the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

FPT_RCV.3.2 For [**loss of power or reset not during database operations**], the TSF shall ensure the return of the TOE to a secure state using automated procedures.

FPT_RCV.3.3 The functions provided by the TSF to recover from failure or service discontinuity shall ensure that the secure initial state is restored without exceeding [**3 database records**] for loss of TSF data or objects within the TSC.

FPT_RCV.3.4 The TSF shall provide the capability to determine the objects that were or were not capable of being recovered.

5.1.5.4 Function recovery (FPT_RCV.4)

FPT_RCV.4.1 The TSF shall ensure that [**all security functions recover to a consistent and secure state after a SNS power outage or reset**] have the property that the SF either completes successfully, or for the indicated failure scenarios, recovers to a consistent and secure state.

5.1.5.5 Non-bypassability of the TSP (FPT_RVM.1)

FPT_RVM.1.1 The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

5.1.5.6 Complete reference monitor (FPT_SEP.3)

FPT_SEP.3.1 The unisolated portion of the TSF shall maintain a security domain for its own execution that protects it from interference and tampering by least privilege subjects.

FPT_SEP.3.2 The TSF shall enforce separation between the security domains of subjects in the TSC.

FPT_SEP.3.3 The TSF shall maintain the part of the TSF that enforces the access control and/or information flow control SFPs in a security domain for its own execution that protects them from interference and tampering by the remainder of the TSF and by subjects least privilege with respect to the TSP.

5.1.5.7 Reliable time stamps (FPT_STM.1)

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps for its own use.

5.1.5.8 Inter-TSF basic TSF data consistency (FPT_TDC.1)

FPT_TDC.1.1 The TSF shall provide the capability to consistently interpret [**sensitivity labels**] when shared between the TSF and another trusted IT product.

FPT_TDC.1.2 The TSF shall use [**CIPSO Domain of Interpretation rules**] when interpreting the TSF data from another trusted IT product.

5.1.5.9 TSF testing (FPT_TST.1)

FPT_TST.1.1 The TSF shall run a suite of self tests [***during initial start-up***] to demonstrate the correct operation of [***[interval timer, memory, PIC]***].

FPT_TST.1.2 The TSF shall provide authorised users with the capability to verify the integrity of [***SNS Database***].

FPT_TST.1.3 The TSF shall provide authorised users with the capability to verify the integrity of stored TSF executable code.

5.1.6 Resource utilization (FRU)

5.1.6.1 Limited fault tolerance (FRU_FLT.2)

FRU_FLT.2.1 The TSF shall ensure the operation of all the TOE's capabilities when the following failures occur: [**failure of non-NM peers or peer-to-peer communications**].

5.1.6.2 Maximum quotas (FRU_RSA.1)

FRU_RSA.1.1 The TSF shall enforce maximum quotas of the following resources: [***datagram throughput***] that [***the traffic through an interface***] can use [***over a specified period of time***].

5.1.7 TOE access (FTA)

5.1.7.1 TOE access history (FTA_TAH.1)

FTA_TAH.1.1 Upon successful session establishment, the TSF shall display the [***date*** **and** ***time***] of the last successful session establishment to the user.

FTA_TAH.1.2 Upon successful session establishment, the TSF shall display the [***date*** **and** ***time***] of the last unsuccessful attempt to session establishment and the number of unsuccessful attempts since the last successful session establishment.

FTA_TAH.1.3 The TSF shall not erase the access history information from the user interface without giving the user an opportunity to review the information.

5.2 TOE Security Assurance Requirements

The security assurance requirements for the TOE are EAL 5 augmented with ACM_AUT.2, ACM_CAP.5, ADO_DEL.3, ADV_HLD.4, ADV_IMP.3, ADV_INT.3, ADV_LLD.2, ADV_RCR.3, ALC_DVS.2, ALC_FLR.2, ALC_LCD.3, ALC_TAT.3, ATE_COV.3, ATE_DPT.3, ATE_FUN.2, AVA_CCA.2, and AVA_MSU.3 components as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components.

Requirement Class	Requirement Component
ACM: Configuration management	ACM_AUT.2: Complete CM automation
	ACM_CAP.5: Advanced support
	ACM_SCP.3: Development tools CM coverage
ADO: Delivery and operation	ADO_DEL.3: Prevention of modification
	ADO_IGS.1: Installation, generation, and start-up procedures
ADV: Development	ADV_FSP.3: Semiformal functional specification
	ADV_HLD.4: Semiformal high-level design
	ADV_IMP.3: Structured implementation of the TSF
	ADV_INT.3: Minimisation of complexity
	ADV_LLD.2: Semiformal low-level design
	ADV_RCR.3: Formal correspondence demonstration
	ADV_SPM.3: Formal TOE security policy model
AGD: Guidance documents	AGD_ADM.1: Administrator guidance
	AGD_USR.1: User guidance
ALC: Life cycle support	ALC_DVS.2: Sufficiency of security measures
	ALC_FLR.2: Flaw reporting procedures
	ALC_LCD.3: Measurable life-cycle model
	ALC_TAT.3: Compliance with implementation standards - all parts
ATE: Tests	ATE_COV.3: Rigorous analysis of coverage
	ATE_DPT.3: Testing: implementation representation
	ATE_FUN.2: Ordered functional testing
	ATE_IND.2: Independent testing - sample
AVA: Vulnerability assessment	AVA_CCA.2: Systematic covert channel analysis
	AVA_MSU.3: Analysis and testing for insecure states
	AVA_SOF.1: Strength of TOE security function evaluation
	AVA_VLA.3: Moderately resistant

Table 2 EAL 5 augmented with ACM_AUT.2, ACM_CAP.5, ADO_DEL.3, ADV_HLD.4, ADV_IMP.3, ADV_INT.3, ADV_LLD.2, ADV_RCR.3, ALC_DVS.2, ALC_FLR.2, ALC_LCD.3, ALC_TAT.3, ATE_COV.3, ATE_DPT.3, ATE_FUN.2, AVA_CCA.2, and AVA_MSU.3 Assurance Components

5.2.1 Configuration management (ACM)

5.2.1.1 Complete CM automation (ACM_AUT.2)

ACM_AUT.2.1d The developer shall use a CM system.
ACM_AUT.2.2d The developer shall provide a CM plan.
ACM_AUT.2.1c The CM system shall provide an automated means by which only authorized changes are made to the TOE implementation representation, and to all other configuration items.
ACM_AUT.2.2c The CM system shall provide an automated means to support the generation of the TOE.
ACM_AUT.2.3c The CM plan shall describe the automated tools used in the CM system.
ACM_AUT.2.4c The CM plan shall describe how the automated tools are used in the CM system.
ACM_AUT.2.5c The CM system shall provide an automated means to ascertain the changes between the TOE and its preceding version.
ACM_AUT.2.6c The CM system shall provide an automated means to identify all other configuration items that are affected by the modification of a given configuration item.
ACM_AUT.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.1.2 Advanced support (ACM_CAP.5)

ACM_CAP.5.1d The developer shall provide a reference for the TOE.
ACM_CAP.5.2d The developer shall use a CM system.
ACM_CAP.5.3d The developer shall provide CM documentation.
ACM_CAP.5.1c The reference for the TOE shall be unique to each version of the TOE.
ACM_CAP.5.2c The TOE shall be labeled with its reference.
ACM_CAP.5.3c The CM documentation shall include a configuration list, a CM plan, an acceptance plan, and integration procedures.
ACM_CAP.5.4c The configuration list shall describe the configuration items that comprise the TOE.
ACM_CAP.5.5c The configuration list shall uniquely identify all configuration items that comprise the TOE.
ACM_CAP.5.6c The CM documentation shall describe the method used to uniquely identify the configuration items that comprise the TOE.
ACM_CAP.5.7c The CM system shall uniquely identify all configuration items that comprise the TOE.
ACM_CAP.5.8c The CM plan shall describe how the CM system is used.
ACM_CAP.5.9c The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.
ACM_CAP.5.10c The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.
ACM_CAP.5.11c The CM system shall provide measures such that only authorized changes are made to the configuration items.
ACM_CAP.5.12c The CM system shall support the generation of the TOE.
ACM_CAP.5.13c The acceptance plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.
ACM_CAP.5.14c The integration procedures shall describe how the CM system is applied in the TOE manufacturing process.
ACM_CAP.5.15c The CM system shall require that the person responsible for accepting a configuration item into CM is not the person who developed it.
ACM_CAP.5.16c The CM system shall clearly identify the configuration items that comprise the TSF.
ACM_CAP.5.17c The CM system shall support the audit of all modifications to the TOE, including the originator, date, and time in the audit trail.
ACM_CAP.5.18c The CM system shall be able to identify the master copy of all material used to generate the TOE.
ACM_CAP.5.19c The CM documentation shall demonstrate that the use of the CM system, together with the development security measures, allow only authorized changes to be made to the TOE.
ACM_CAP.5.20c The CM documentation shall demonstrate that the use of the integration procedures ensures that the generation of the TOE is correctly performed in an authorized manner.
ACM_CAP.5.21c The CM documentation shall demonstrate that the CM system is sufficient to ensure that the person responsible for accepting a configuration item into CM is not the person who developed it.

ACM_CAP.5.22c The CM documentation shall justify that the acceptance procedures provide for an adequate and appropriate review of changes to all configuration items.
ACM_CAP.5.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.1.3 Development tools CM coverage (ACM_SCP.3)

ACM_SCP.3.1d The developer shall provide a list of configuration items for the TOE.
ACM_SCP.3.1c The list of configuration items shall include the following: implementation representation; security flaws; development tools and related information; and the evaluation evidence required by the assurance components in the ST.
ACM_SCP.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.2 Delivery and operation (ADO)

5.2.2.1 Prevention of modification (ADO_DEL.3)

ADO_DEL.3.1d The developer shall document procedures for delivery of the TOE or parts of it to the user.
ADO_DEL.3.2d The developer shall use the delivery procedures.
ADO_DEL.3.1c The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.
ADO_DEL.3.2c The delivery documentation shall describe how the various procedures and technical measures provide for the prevention of modifications, or any discrepancy between the developer's master copy and the version received at the user site.
ADO_DEL.3.3c The delivery documentation shall describe how the various procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.
ADO_DEL.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.2.2 Installation, generation, and start-up procedures (ADO_IGS.1)

ADO_IGS.1.1d The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.
ADO_IGS.1.1c The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation and start-up of the TOE.
ADO_IGS.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADO_IGS.1.2e The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

5.2.3 Development (ADV)

5.2.3.1 Semiformal functional specification (ADV_FSP.3)

ADV_FSP.3.1d The developer shall provide a functional specification.
ADV_FSP.3.1c The functional specification shall describe the TSF and its external interfaces using **a semiformal style, supported by** informal, **explanatory text where appropriate.**
ADV_FSP.3.2c The functional specification shall be internally consistent.
ADV_FSP.3.3c The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing complete details of all effects, exceptions and error messages.
ADV_FSP.3.4c The functional specification shall completely represent the TSF.
ADV_FSP.3.5c The functional specification shall include rationale that the TSF is completely represented.
ADV_FSP.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.3.2e The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

Note that the SNS separation kernel and reference monitor infrastructure, including that for IP routing and MLS policy enforcement, were specified and evaluated formally per ADV_FSP.4 and that the SNS functional specification and evidence package reflect this. The SNS filter engine was specified and evaluated semi-formally per ADV_FSP.3. As a result, the SNS in its entirety meets ADV_FSP.3.

5.2.3.2 Semiformal high-level design (ADV_HLD.4)

ADV_HLD.4.1d The developer shall provide the high-level design of the TSF.
ADV_HLD.4.1c The presentation of the high-level design shall be semiformal.
ADV_HLD.4.2c The high-level design shall be internally consistent.
ADV_HLD.4.3c The high-level design shall describe the structure of the TSF in terms of subsystems.
ADV_HLD.4.4c The high-level design shall describe the security functionality provided by each subsystem of the TSF.
ADV_HLD.4.5c The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.
ADV_HLD.4.6c The high-level design shall identify all interfaces to the subsystems of the TSF.
ADV_HLD.4.7c The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.
ADV_HLD.4.8c The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing complete details of all effects, exceptions and error messages.
ADV_HLD.4.9c The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.
ADV_HLD.4.10c The high-level design shall justify that the identified means of achieving separation, including any protection mechanisms, are sufficient to ensure a clear and effective separation of TSP-enforcing from non-TSP-enforcing functions.
ADV_HLD.4.11c The high-level design shall justify that the TSF mechanisms are sufficient to implement the security functions identified in the high-level design.
ADV_HLD.4.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_HLD.4.2e The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

Note that the high-level design of the SNS separation kernel and reference monitor infrastructure, including that for IP routing and MLS policy, was specified and evaluated formally per ADV_HLD.5 and that the SNS specifications, formal models, and evidence package reflect this. The high-level design of the SNS filter engine was specified and evaluated semi-formally. As a result, the SNS in its entirety meets ADV_HLD.4.

5.2.3.3 Structured implementation of the TSF (ADV_IMP.3)

ADV_IMP.3.1d The developer shall provide the implementation representation for the entire TSF.
ADV_IMP.3.1c The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.
ADV_IMP.3.2c The implementation representation shall be internally consistent.
ADV_IMP.3.3c The implementation representation shall describe the relationships between all portions of the implementation.
ADV_IMP.3.4c The implementation representation shall be structured into small and comprehensible sections.
ADV_IMP.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_IMP.3.2e The evaluator shall determine that the implementation representation is an accurate and complete instantiation of the TOE security functional requirements.

5.2.3.4 Minimisation of complexity (ADV_INT.3)

ADV_INT.3.1d The developer shall design and structure the TSF in a modular fashion that avoids unnecessary interactions between the modules of the design.
ADV_INT.3.2d The developer shall provide an architectural description.
ADV_INT.3.3d The developer shall design and structure the TSF in a layered fashion that minimizes mutual interactions between the layers of the design.
ADV_INT.3.4d The developer shall design and structure the TSF in such a way that minimizes the complexity of the entire TSF.
ADV_INT.3.5d The developer shall design and structure the portions of the TSF that enforce any access control and/or information flow control policies such that they are simple enough to be analyzed.
ADV_INT.3.6d The developer shall ensure that functions whose objectives are not relevant for the TSF are excluded from the TSF modules.
ADV_INT.3.1c The architectural description shall identify the modules of the TSF and shall specify which portions of the TSF enforce the access control and/or information flow control policies.
ADV_INT.3.2c The architectural description shall describe the purpose, interface, parameters, and side-effects of each module of the TSF.
ADV_INT.3.3c The architectural description shall describe how the TSF design provides for largely independent modules that avoid unnecessary interactions.
ADV_INT.3.4c The architectural description shall describe the layering architecture.
ADV_INT.3.5c The architectural description shall show that mutual interactions have been minimized, and justify those that remain.
ADV_INT.3.6c The architectural description shall describe how the entire TSF has been structured to minimize complexity.
ADV_INT.3.7c The architectural description shall justify the inclusion of any non-TSP-enforcing modules in the TSF.
ADV_INT.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_INT.3.2e The evaluator shall determine that both the low-level design and the implementation representation are in compliance with the architectural description.
ADV_INT.3.3e The evaluator shall confirm that the portions of the TSF that enforce any access control and/or information flow control policies are simple enough to be analyzed.

5.2.3.5 Semiformal low-level design (ADV_LLD.2)

ADV_LLD.2.1d The developer shall provide the low-level design of the TSF.
ADV_LLD.2.1c The presentation of the low-level design shall be semiformal.
ADV_LLD.2.2c The low-level design shall be internally consistent.
ADV_LLD.2.3c The low-level design shall describe the TSF in terms of modules.
ADV_LLD.2.4c The low-level design shall describe the purpose of each module.
ADV_LLD.2.5c The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.
ADV_LLD.2.6c The low-level design shall describe how each TSP-enforcing function is provided.
ADV_LLD.2.7c The low-level design shall identify all interfaces to the modules of the TSF.
ADV_LLD.2.8c The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.
ADV_LLD.2.9c The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing complete details of all effects, exceptions and error messages.
ADV_LLD.2.10c The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.
ADV_LLD.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_LLD.2.2e The evaluator shall determine that the low-level design is an accurate and complete instantiation of the TOE security functional requirements.

5.2.3.6 Semiformal correspondence demonstration (ADV_RCR.3)

ADV_RCR.3.1d The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.
ADV_RCR.3.1c For those corresponding portions of representations that are formally specified, the developer shall prove that correspondence.
ADV_RCR.3.2c For each adjacent pair of provided TSF representations, the analysis shall prove or demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.
ADV_RCR.3.3c For each adjacent pair of provided TSF representations, where portions of one representation are semiformally specified and the other at least semiformally specified, the demonstration of correspondence between those portions of the representations shall be semiformal.
ADV_RCR.3.4c For each adjacent pair of provided TSF representations, where portions of both representations are formally specified, the proof of correspondence between those portions of the representations shall be formal.
ADV_RCR.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_RCR.3.2e The evaluator shall determine the accuracy of the proofs of correspondence by selectively verifying the formal analysis.

5.2.3.7 Formal TOE security policy model (ADV_SPM.3)

ADV_SPM.3.1d The developer shall provide a TSP model.
ADV_SPM.3.2d The developer shall demonstrate or prove, as appropriate, correspondence between the functional specification and the TSP model.
ADV_SPM.3.1c The TSP model shall be formal.
ADV_SPM.3.2c The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.
ADV_SPM.3.3c The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.
ADV_SPM.3.4c The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.
ADV_SPM.3.5c Where the functional specification is semiformal, the demonstration of correspondence between the TSP model and the functional specification shall be semiformal.
ADV_SPM.3.6c Where the functional specification is formal, the proof of correspondence between the TSP model and the functional specification shall be formal.
ADV_SPM.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.4 Guidance documents (AGD)

5.2.4.1 Administrator guidance (AGD_ADM.1)

AGD_ADM.1.1d The developer shall provide administrator guidance addressed to system administrative personnel.
AGD_ADM.1.1c The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.
AGD_ADM.1.2c The administrator guidance shall describe how to administer the TOE in a secure manner.
AGD_ADM.1.3c The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.
AGD_ADM.1.4c The administrator guidance shall describe all assumptions regarding user behaviour that are relevant to secure operation of the TOE.
AGD_ADM.1.5c The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.
AGD_ADM.1.6c The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_ADM.1.7c The administrator guidance shall be consistent with all other documentation supplied for evaluation.
AGD_ADM.1.8c The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.
AGD_ADM.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.4.2 User guidance (AGD_USR.1)

AGD_USR.1.1d The developer shall provide user guidance.
AGD_USR.1.1c The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.
AGD_USR.1.2c The user guidance shall describe the use of user-accessible security functions provided by the TOE.
AGD_USR.1.3c The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.
AGD_USR.1.4c The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behaviour found in the statement of TOE security environment.
AGD_USR.1.5c The user guidance shall be consistent with all other documentation supplied for evaluation.
AGD_USR.1.6c The user guidance shall describe all security requirements for the IT environment that are relevant to the user.
AGD_USR.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.5 Life cycle support (ALC)

5.2.5.1 Sufficiency of security measures (ALC_DVS.2)

ALC_DVS.2.1d The developer shall produce development security documentation.
ALC_DVS.2.1c The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.
ALC_DVS.2.2c The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.
ALC_DVS.2.3c The evidence shall justify that the security measures provide the necessary level of protection to maintain the confidentiality and integrity of the TOE.
ALC_DVS.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ALC_DVS.2.2e The evaluator shall confirm that the security measures are being applied.

5.2.5.2 Flaw reporting procedures (ALC_FLR.2)

ALC_FLR.2.1d The developer shall provide flaw remediation procedures addressed to TOE developers.
ALC_FLR.2.2d The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.
ALC_FLR.2.3d The developer shall provide flaw remediation guidance addressed to TOE users.
ALC_FLR.2.1c The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.
ALC_FLR.2.2c The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.
ALC_FLR.2.3c The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.
ALC_FLR.2.4c The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.
ALC_FLR.2.5c The flaw remediation procedures documentation shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.

ALC_FLR.2.6c The procedures for processing reported security flaws shall ensure that any reported flaws are corrected and the correction issued to TOE users.

ALC_FLR.2.7c The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.

ALC_FLR.2.8c The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.

ALC_FLR.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.5.3 Measurable life-cycle model (ALC_LCD.3)

ALC_LCD.3.1d The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC_LCD.3.2d The developer shall provide life-cycle definition documentation.

ALC_LCD.3.3d The developer shall use a standardised and measurable life-cycle model to develop and maintain the TOE.

ALC_LCD.3.4d The developer shall measure the TOE development using the standardised and measurable life-cycle model.

ALC_LCD.3.1c The life-cycle definition documentation shall describe the model used to develop and maintain the TOE, including the details of its arithmetic parameters and/or metrics used to measure the TOE development against the model.

ALC_LCD.3.2c The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.

ALC_LCD.3.3c The life-cycle definition documentation shall explain why the model was chosen.

ALC_LCD.3.4c The life-cycle definition documentation shall explain how the model is used to develop and maintain the TOE.

ALC_LCD.3.5c The life-cycle definition documentation shall demonstrate compliance with the standardised and measurable life-cycle model.

ALC_LCD.3.6c The life-cycle documentation shall provide the results of the measurements of the TOE development using the standardised and measurable life-cycle model.

ALC_LCD.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.5.4 Compliance with implementation standards - all parts (ALC_TAT.3)

ALC_TAT.3.1d The developer shall identify the development tools being used for the TOE.

ALC_TAT.3.2d The developer shall document the selected implementation-dependent options of the development tools.

ALC_TAT.3.3d The developer shall describe the implementation standards for all parts of the TOE.

ALC_TAT.3.1c All development tools used for implementation shall be well-defined.

ALC_TAT.3.2c The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.

ALC_TAT.3.3c The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options.

ALC_TAT.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_TAT.3.2e The evaluator shall confirm that the implementation standards have been applied.

5.2.6 Tests (ATE)

5.2.6.1 Rigorous analysis of coverage (ATE_COV.3)

ATE_COV.3.1d The developer shall provide an analysis of the test coverage.

ATE_COV.3.1c The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

ATE_COV.3.2c The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.
ATE_COV.3.3c The analysis of the test coverage shall rigorously demonstrate that all external interfaces of the TSF identified in the functional specification have been completely tested.
ATE_COV.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.6.2 Testing: implementation representation (ATE_DPT.3)

ATE_DPT.3.1d The developer shall provide the analysis of the depth of testing.
ATE_DPT.3.1c The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design, low-level design and implementation representation.
ATE_DPT.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.6.3 Ordered functional testing (ATE_FUN.2)

ATE_FUN.2.1d The developer shall test the TSF and document the results.
ATE_FUN.2.2d The developer shall provide test documentation.
ATE_FUN.2.1c The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.
ATE_FUN.2.2c The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.
ATE_FUN.2.3c The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.
ATE_FUN.2.4c The expected test results shall show the anticipated outputs from a successful execution of the tests.
ATE_FUN.2.5c The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.
ATE_FUN.2.6c The test documentation shall include an analysis of the test procedure ordering dependencies.
ATE_FUN.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.6.4 Independent testing - sample (ATE_IND.2)

ATE_IND.2.1d The developer shall provide the TOE for testing.
ATE_IND.2.1c The TOE shall be suitable for testing.
ATE_IND.2.2c The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.
ATE_IND.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ATE_IND.2.2e The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified.
ATE_IND.2.3e The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

5.2.7 Vulnerability assessment (AVA)

5.2.7.1 Systematic covert channel analysis (AVA_CCA.2)

AVA_CCA.2.1d The developer shall conduct a search for covert channels for each information flow control policy.
AVA_CCA.2.2d The developer shall provide covert channel analysis documentation.
AVA_CCA.2.1c The analysis documentation shall identify covert channels and estimate their capacity.
AVA_CCA.2.2c The analysis documentation shall describe the procedures used for determining the existence of covert channels, and the information needed to carry out the covert channel analysis.

AVA_CCA.2.3c The analysis documentation shall describe all assumptions made during the covert channel analysis.
AVA_CCA.2.4c The analysis documentation shall describe the method used for estimating channel capacity, based on worst case scenarios.
AVA_CCA.2.5c The analysis documentation shall describe the worst case exploitation scenario for each identified covert channel.
AVA_CCA.2.6c The analysis documentation shall provide evidence that the method used to identify covert channels is systematic.
AVA_CCA.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_CCA.2.2e The evaluator shall confirm that the results of the covert channel analysis show that the TOE meets its functional requirements.
AVA_CCA.2.3e The evaluator shall selectively validate the covert channel analysis through testing.

5.2.7.2 Analysis and testing for insecure states (AVA_MSU.3)

AVA_MSU.3.1d The developer shall provide guidance documentation.
AVA_MSU.3.2d The developer shall document an analysis of the guidance documentation.
AVA_MSU.3.1c The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.
AVA_MSU.3.2c The guidance documentation shall be complete, clear, consistent and reasonable.
AVA_MSU.3.3c The guidance documentation shall list all assumptions about the intended environment.
AVA_MSU.3.4c The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).
AVA_MSU.3.5c The analysis documentation shall demonstrate that the guidance documentation is complete.
AVA_MSU.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_MSU.3.2e The evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.
AVA_MSU.3.3e The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.
AVA_MSU.3.4e The evaluator shall confirm that the analysis documentation shows that guidance is provided for secure operation in all modes of operation of the TOE.
AVA_MSU.3.5e The evaluator shall perform independent testing to determine that an administrator or user, with an understanding of the guidance documentation, would reasonably be able to determine if the TOE is configured and operating in a manner that is insecure.

5.2.7.3 Strength of TOE security function evaluation (AVA_SOF.1)

AVA_SOF.1.1d The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.
AVA_SOF.1.1c For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.
AVA_SOF.1.2c For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.
AVA_SOF.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_SOF.1.2e The evaluator shall confirm that the strength claims are correct.

5.2.7.4 Moderately resistant (AVA_VLA.3)

AVA_VLA.3.1d The developer shall perform a vulnerability analysis.

AVA_VLA.3.2d The developer shall provide vulnerability analysis documentation.

AVA_VLA.3.1c The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for ways in which a user can violate the TSP.

AVA_VLA.3.2c The vulnerability analysis documentation shall describe the disposition of identified vulnerabilities.

AVA_VLA.3.3c The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA_VLA.3.4c The vulnerability analysis documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.

AVA_VLA.3.5c The vulnerability analysis documentation shall show that the search for vulnerabilities is systematic.

AVA_VLA.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VLA.3.2e The evaluator ***shall conduct*** penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.

AVA_VLA.3.3e The evaluator ***shall perform*** an independent vulnerability analysis.

AVA_VLA.3.4e The evaluator ***shall perform*** independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of additional identified vulnerabilities in the intended environment.

AVA_VLA.3.5e The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker possessing a **moderate** attack potential.

6. TOE Summary Specification

This chapter describes the security functions and associated assurance measures.

6.1 TOE Security Functions

6.1.1 Security audit

The TOE is designed to audit security relevant events as they occur and record those events in an event log. All of the audit events are forwarded from distributed SNSs to the Network Management (NM) node using RDP. The NM protects the event log so that only the SA and SSA can review and otherwise manage (e.g., archive) the audit events. This is accomplished by displaying audit records only on the SA console and offering audit review commands only via the SA console once an SA or SSA is logged in.

The NM prints warnings (i.e., alarms) to the SA console (whether the SA/SSA is logged in or not) when potential security violations may be indicated and when the available audit storage space is becoming exhausted. It also provides the ability to configure the NM to either shutdown or to begin overwriting the oldest audit records should exhaustion occur.

The TOE can be configured to filter the generated audit events based on the type of event as well as the sensitivity level involved in the event.

The audit events generated by the SNS correspond to the required events below:

Related SFRs	Auditable Events	Audit Error Messages
FAU_GEN.1	• Start-up and shutdown of audit function	sm_start_if sm_shut_if
FAU_GEN.1	• Use of covert channel related mechanism	sa_audit_set sa_audit_unset
FAU_ARP.1	• Actions taken due to imminent security violations	There is an alarm count for the audit event. Alarm is printed on the SA console. Also, the SNS generates an audit (most of the audits are for 'imminent security violations') and then drop the packet or denies the action.
FAU_SAA.1	• Enabling and disabling of any of the analysis mechanisms • Automated responses performed by the tool	sa_audit_set sa_audit_unset Also see FAU_ARP.1
FAU_SAR.1	• Reading of information from the audit records	sa_display_audit
FAU_SAR.2	• Unsuccessful attempts to read information from the audit records	An unsuccessful attempt will result in a HALT.
FAU_SEL.1	• All modifications to the audit configuration that occur while the audit collection functions are operating	sa_audit_filter
FAU_STG.3	• Actions taken due to exceeding of a threshold	NM_NET_AUDIT_FILE - xx of 100000000 used - archive soon

Related SFRs	Auditable Events	Audit Error Messages
FAU_STG.4	• Actions taken due to the audit storage failure	NM_NET_AUDIT_FILE - ARCHIVE IMMEDIATELY
FDP_ETC.1 FDP_ETC.2	• All attempts to export information	ip_send_sens_range ip_tcp_sens_range ip_writeup_sens_range ip_writeup_forbidden ip_writeup_impossible ip_downgrade_forbidden
FDP_IFF.2	• All decisions on requests for information flow	ip_audit_sens ip_writeup_sens_range ip_writeup_forbidden ip_writeup_impossible ip_downgrade_forbidden ip_tcp_sens_range ip_receive_sens_range ip_send_sens_range ip_receive_dac ip_receive_ntcb_tag
FDP_IFF.4	• All decisions on requests for information flow • The use of identified illicit information flow channels (e.g.g, file system or task group related events that may indicate attempted covert channel use)	ip_audit_sens ip_writeup_sens_range ip_writeup_forbidden ip_writeup_impossible ip_downgrade_forbidden ip_tcp_sens_range ip_receive_sens_range ip_send_sens_range ip_receive_dac ip_receive_ntcb_tag ip_open_fail ip_delete_fail ip_no_write_access ip_open_fail_quota ip_open_fail_file_corrupt ip_tcp_sens_range ip_resource_error ip_cant_create_task_grp ip_cant_bind_to_socket

Related SFRs	Auditable Events	Audit Error Messages
		ip_fs_exceptions
FDP_ITC.1 FDP_ITC.2	• All attempts to import user data, including any security attributes	ip_receive_sens_range ip_receive_dac ip_receive_ntcb_tag ip_tcp_sens_range
FIA_AFL.1	• Reaching of the threshold for the unsuccessful authentication attempts and the actions (e.g. disabling of a terminal) taken and the subsequent, if appropriate, restoration to the normal state (e.g. re-enabling of a terminal)	na_login_failure sa_login_failure uty_login_failure The alarm limit setting for the audit event. Inside the audit event record will be a reason for the failure, such as “temporarily barred” which in this case represents the user exceeded the login threshold (set by SA).
FIA_SOS.1	• Rejection or acceptance by the TSF of any tested secret	na_passwd_change_failure na_password_cmd sa_passwd_change_failure uty_passwd_change_failure
FIA_UAU.1	• All use of the authentication mechanism	na_login_failure na_login sa_login_failure sa_user_failed_login sa_login uty_login_failure uty_login
FIA_UID.2	• All use of the user identification mechanism, including the user identity provided	na_login_failure na_login sa_login_failure sa_user_failed_login sa_login uty_login_failure uty_login
FMT_MOF.1	• All modifications in the behaviour of the functions in the TSF	na_dac_add na_dac_delete na_device_add na_device_delete na_device_modify na_display_dac na_monitor_alarm_reset

Related SFRs	Auditable Events	Audit Error Messages
		na_monitor_alarm_display
		na_monitor_set
		na_monitor_unset
		sa_passwd_change_failure
		sa_admin_delete_failure
		sa_admin_modify_failure
		sa_audit_filter
		sa_audit_overwrite
		sa_audit_alarm_reset
		sa_audit_alarm_display
		sa_audit_set
		sa_audit_unset
		sa_device_modify
		sa_display_audit
		sa_display_device
		sa_display_user
		sa_label_assign
		sa_label_reassign
		sa_label_unassign
		sa_list_audit
		sa_list_device
		sa_list_label
		sa_list_user
		sa_restart_cmd
		sa_shutdown_cmd
		sa_user_add
		sa_user_delete
		sa_user_modify
		sa_user_name_length
		sa_user_password_length
		sa_user_login_delay
		sa_user_failed_login
		sa_user_pwd_complexity
		sa_time_cmd
		sa_proxy_add
		sa_proxy_delete
		sa_threat_level

Related SFRs	Auditable Events	Audit Error Messages
		sa_filter_file_remove sa_display_filter_file sa_review_filter_file sa_list_filter_files sa_filter_file_remove_fail sa_filter_file_update sa_review_rules_file uty_passwd_change_failure uty_erase_file
FMT_MSA.1	• All modifications of the values of security attributes	sa_label_assign sa_label_reassign sa_label_unassign
FMT_MSA.3	• Modifications of the default setting of permissive or restrictive rules • All modifications of the initial values of security attributes	na_device_add na_device_modify sa_device_modify
FMT_MTD.1	• All modifications to the values of TSF data	sa_admin_delete_failure sa_admin_modify_failure sa_user_add
FMT_SAE.1	• Specification of the expiration time for an attribute • Action taken due to attribute expiration	sa_user_add sa_user_modify na_login_failure sa_login_failure uty_login_failure
FMT_SMF.1	• Use of the management functions	na_dac_add na_dac_delete na_device_add na_device_delete na_device_modify na_display_dac na_monitor_alarm_reset na_monitor_alarm_display na_monitor_set na_monitor_unset na_operational_cmd na_password_cmd na_router_add

Related SFRs	Auditable Events	Audit Error Messages
		na_router_delete na_shutdown_cmd na_time_cmd sa_passwd_change_failure sa_admin_delete_failure sa_admin_modify_failure sa_audit_filter sa_audit_overwrite sa_audit_alarm_reset sa_audit_alarm_display sa_audit_set sa_audit_unset sa_device_modify sa_display_audit sa_display_device sa_display_user sa_label_assign sa_label_reassign sa_label_unassign sa_list_audit sa_list_device sa_list_label sa_list_user sa_restart_cmd sa_shutdown_cmd sa_user_add sa_user_delete sa_user_modify sa_user_name_length sa_user_password_length sa_user_login_delay sa_user_failed_login sa_user_pwd_complexity sa_time_cmd sa_proxy_add sa_proxy_delete sa_threat_level

Related SFRs	Auditable Events	Audit Error Messages
		sa_filter_file_remove sa_display_filter_file sa_review_filter_file sa_list_filter_files sa_filter_file_remove_fail sa_filter_file_update sa_review_rules_file uty_passwd_change_failure uty_erase_file
FMT_SMR.2	• Modifications to the group of users that are part of a role • Unsuccessful attempts to use a role due to the given conditions on the roles	sa_user_add sa_user_delete sa_user_modify na_login_failure sa_login_failure uty_login_failure
FMT_SMR.3	• Explicit request to assume a role	na_login sa_login uty_login
FPT_FLS.1	• Failure of the TSF	All halts represent fatal failure. No audit for such event since the SNS does not take the time to continue disk access before halting the CPU.
FPT_RCV.3	• Resumption of the regular operation • Type of failure or service discontinuity	sa_hw_reset na_hw_reset na_operational_cmd
FPT_RCV.4	• The detection of a failure of a security function	sa_hw_reset na_hw_reset
FPT_STM.1	• Changes to the time	na_time_cmd sa_time_cmd
FPT_TDC.1	• Use of the TSF data consistency mechanisms • Identification of which TSF data have been interpreted • Detection of modified TSF data	ip_tcp_sens_range ip_receive_sens_range ip_send_sens_range
FRU_FLT.2	• Any failure detected by the TSF	ip_proxy_session_failure na_sns_unreachable sa_sns_unreachable
FRU_RSA.1	• All attempted uses of the resource allocation functions for resources that are	na_device_add

Related SFRs	Auditable Events	Audit Error Messages
	under control of the TSF	na_device_modify

The Security audit function is designed to satisfy the following security functional requirements:

- FAU_ARP.1: The SNS generates an alarm when an accumulation of alarm events hits an SA/SSA-specified threshold within a SA/SSA-specified time duration. The alarm is printed on the SA console, the system denies the current access attempt, and for covert storage channel audits, blocks further (by not acknowledging receipt of the audit event causing the action to be suspended) action until the SA/SSA-specified time duration has expired.
- FAU_GEN.1: The SNS generates the audit events indicated above, recording the data and time, event type, applicable subject identity and outcome. The Trusted Facilities Manual, Appendix D.1, lists all audit events generated by the SNS.
- FAU_SAA.1: The SNS monitors accumulation of events that indicate an attempt to violate policy. This involves, for a given event type, counting failed events occurring between successful events. For audits related to covert channels, the SNS enforces a policy of limiting the number of events per time period (both set by the Security Administrator) by blocking further events by the offending subject until the time period expires. For each individual audit event, if the number of events exceeds a threshold in an optional time period, an alarm is issued.
- FAU_SAR.1: The SA and SSA are permitted read access to audit trails via interfaces accessible from the SA console. The SNS provides text display of audit trail records.
- FAU_SAR.2: Only the SA and SSA are permitted read access to audit trails since they are accessible only via the SA console after the SA/SSA logs in.
- FAU_SEL.1: The SNS pre-filters auditing based on event types and sensitivity levels selected by the SA and SSA.
- FAU_STG.2: Only the SA and SSA are provided interfaces to delete records and they will succeed only if either the records have been archived or the system has audit overwrite enabled by the SA or SSA. The SNS prevents unauthorized (in fact all) modification to audit records by simply offering no interfaces to perform this operation. In non-overwrite mode, the SNS blocks auditable events (by not acknowledging the receipt of audit events which serves to suspend actions) when audit trail is full. Once the audit trail is archived and erased, events that have not yet been written into the audit trail are saved, and the events are acknowledged allowing the associated actions to resume. In this manner, no audit records are lost when the audit trail becomes full.
- FAU_STG.3: The SNS periodically warns the SA/SSA beginning when the audit trail is 90% full, and the frequency of warning increases as the trail becomes fuller.
- FAU_STG.4: In non-overwrite mode, the SNS blocks auditable events (by not acknowledging audit records causing the action to be suspended) when the audit trail is full. Once the audit trail is archived and erased, events waiting to be written to the audit trail are saved and audit records are acknowledged allowing the associated actions to resume

6.1.2 User data protection

The TOE is designed to control the flow of information among attached subscriber devices subject to a number of information flow policies.

Each subscriber device has an associated sensitivity label range, which could be degenerate (i.e., a single label). When labeled datagrams are received from a given subscriber device, the SNS requires that the sensitivity label is within the range defined for that subscriber. When unlabeled datagrams are received, they are assigned the default sensitivity label defined for that subscriber device (which must be within the range defined for that subscriber).

Similarly, when datagrams are sent to a subscriber device, the sensitivity label must be within the range defined for that subscriber, and the sensitivity label will be removed from the data if the subscriber device is single-level.

The TOE also supports packet filtering (IP DAC) rules that can be defined by the NA. The rules can define the data of datagrams that will be accepted from subscriber devices based on protocol, source and destination addresses, source and destination ports, and ICMP type. Note that this filtering occurs only on inbound traffic.

Finally, the TOE also supports content filtering that is based on SA/SSA-defined rules. In this case, datagrams being sent between attached subscriber devices are subject to the filter which can approve/disapprove of the traffic or alter the traffic in predefined ways.

Content filtering is based on a set of rules that can be selectively enabled for each port. These rules specify the binary representation of the message content and allow restrictions and transformations.

1. Binary messages
   i. A message will be accepted only if it satisfies a defined structure (based on short, unsigned short, long, unsigned long, byte, unsigned byte, and array fields of each, plus ASCII character arrays), any fields are allowed, any fields have allowed values, message does not contain any defined dirty words, and all character arrays (text fields) are determined to be unlikely to contain text encodings of binary or other data.
   ii. Specified message numerical fields will be zeroed out.
   iii. Specified message textual fields will be replaced with spaces (0x20).

DAC for least privilege proxies is enforced by the use of an “owner” permission assigned to all files that are used by least privilege processes. When a new file is created for a proxy, the TSF assigns the file owner field the proxy’s task ID. This restricts the file from being accessed by any process outside the proxy’s group. When the file is regraded prior to being handed off to the downstream proxy task, the file owner permission is reassigned by the TSF to the downstream proxy task. This denies the previous task further access to the file. It also blocks access to the file by proxies at the same level that are not in the downstream taskgroup.

Least privilege processes are created to handle inbound and outbound packets. These least privilege proxies are outside the TSF. There are two types of these proxies -- Passive and Active. A passive proxy is created for an inbound connection and is dedicated to that connection. An active proxy is created for an outbound connection and is dedicated to that connection. Processes within the TSF perform and control all data flows between the passive and active proxies. Each proxy (active or passive) is assigned to separate and unique task groups.

Furthermore, the TOE ensures that datagrams passing through cannot contain any inappropriate residual information.

The SNS uses the CIPSO standard to represent sensitivity labels.

The User data protection function is designed to satisfy the following security functional requirements:

- FDP_ETC.1: The sensitivity label will be removed from the data on transmission if the subscriber device is single-level and unlabeled.
- FDP_ETC.2: The SNS enforces multi-level access control policy on data transmitted over labeled interfaces. The SNS labels datagrams using CIPSO standard when transmitting data over labeled interfaces. The CIPSO labels are bound to the exported data (i.e., datagrams) as fields in the IP header.
- FDP_IFC.2: The SNS has two policies: Single-level Subscriber Information Flow Policy and Multiple-level Subscriber Information Flow Policy. Each policy performs the following: (1) packet filtering, (2) MAC, and (3) content filtering. The first two are performed on each datagram entering the SNS. MAC is also performed on datagrams leaving the SNS. The MAC policy allows low to high data transfer and allows other flows if they pass through a proxy connection with a filter. For proxied connections with a filter, the content filtering function is based on an SA/SSA-specified rule set. This rule set specifies content filtering for each message field. Content filtering rules include pass-through of a field, zeroizing of a numeric field, replacing text fields with spaces, and discard of message based on field validation failure. Field validation

can be range checks or discrete lists for fields, text recognition to discriminate the more efficient schemes for encoding data in text fields, and dirty word searching of the entire message. SNS ensures that the security policies apply to all subjects and objects.

- FDP_IFF.2: Subjects are subscriber devices. Packet filtering is based on datagram protocol number, source address, target address, TCP/UDP source and target ports, and ICMP type. MAC is based on the sensitivity label associated with each datagram and this label is checked against the range associated with a given subscriber device upon receipt and transmission. Filtering is based on message content rules (as summarized above). All information flows are assigned standard CIPSO sensitivity labels.
- FDP_IFF.4: The SNS controls covert storage channels through the auditing mechanism that limits how many events can occur per SA/SSA-specified duration. The SNS controls covert timing changes through various mechanisms that disrupt the subject's clocks. For internal SNS subjects (proxy processes), the SNS provides a coarse-granularity clock and uses random delays to disrupt the process from building its own clock. For subscriber devices, the SNS can be configured to randomly delay datagrams.
- FDP_ITC.1: Unlabeled datagrams received over unlabeled interfaces are labeled at the sensitivity level of the interface. The SNS discards datagrams from unlabeled interfaces that have an attached label that is different from the sensitivity level of the interface, rather than ignoring the label.
- FDP_ITC.2: The SNS enforces MAC on inbound datagrams over labeled interface. Unlabeled datagrams from labeled interfaces are discarded. SNS uses the CIPSO label bound to the datagram of imported datagrams. CIPSO labels are bound to datagrams as fields in the IP header. The CIPSO label bound to the datagram must be within the assigned sensitivity range for the interface receiving/sending data to the subscriber device. The domain of interpretation (DOI) field identifies a consistent labeling between distributed components. The SNS discards datagrams imported with a different DOI in the IP header than what is configured for the SNS.
- FDP_RIP.2: When a task group is created, all memory that the task group requires for local storage (including stacks, data segments, and task state segment) is allocated to the task group. All system-allocated data is scrubbed (i.e., set to a predetermined initial state) before the memory is added to the new task group's address space. The SNS also ensures that delivered datagrams contain only the data intended for the connection. For file objects, all data residing in the memory/disk space is cleared during file creation.

6.1.3 Identification and authentication

The TOE identifies each attached subscriber device by its unique physical connection. Administrators are defined by the Super-SA with a user identity, role, and are authenticated using passwords. The passwords are subject to a complexity mechanism limiting the available passwords an administrator can choose. Also, passwords and administrator accounts are subject to expiration definable by the Super-SA.

In order to access any TOE function, each administrator must be successfully identified and authenticated. During authentication, the consoles do not echo passwords to mitigate disclosure. Also, once the administrator is logged on, the TOE displays the last time/date of both successful logins and unsuccessful login attempts along with the total count of unsuccessful login attempts since the last successful login.

The Identification and authentication function is designed to satisfy the following security functional requirements:

- FIA_AFL.1: The SNS detects each unsuccessful authentication attempt. When any unsuccessful authentication event occurs, the SNS audits the event and delays until the authentication attempt may be tried. These delays double from an SA/SSA-set starting value to a maximum of 15 minutes.
- FIA_ATD.1: Only administrator users exist for the SNS. The SNS maintains the user identities, roles and authentication data. Settable parameters during account creation include: user name, password, personal data, account type (role), status (good or barred), account expiration date, logon interval and password change interval.

- FIA_SOS.1: The SNS enforces an SA/SSA-settable password difficulty metric. The SA/SSA can specify that at least 1, 2, 3, or 4 of the following are required for passwords: lower case alpha, upper case alpha, numeric, and non-alpha-numeric. The SA/SSA also sets minimum password length.
- FIA_UAU.1: After TOE startup, all administrator actions require authentication to an appropriate console. Subscriber actions are limited by the information flow policies. These actions include MAC/IP DAC checks on incoming data flows and MAC checks on outgoing data flows.
- FIA_UAU.7: The SNS does not echo passwords entered during logon attempts. If an invalid username or password is entered, the SNS delays for a period of time, and then redraws the login banner. No login failure notice is displayed, only a new login prompt.
- FIA_UID.2: The SNS requires every user to be identified - administrators and network subscribers - prior to offering any security functions. Administrator users must logon to an appropriate console before being allowed to perform any admin actions. Network subscribers must pass MAC/IP DAC checks.
- FTA_TAH.1: The SNS displays date and time of last user login for administrator login. The SNS displays date and time of last failed login and number of successive failed logins at the console since the last successful login.
- FMT_SAE.1: The SNS supports expiration of both passwords and user accounts, configurable by the Super-SA. The SNS forces password change on password expiration following a successful authentication. The SNS locks user account on account expiration and “time between login” expiration. Account expiration by default is set at one year ahead of the account creation.

6.1.4 Security management

The TOE provides functions to manage all of its security features. These functions can only be accessed by appropriate administrators from their corresponding consoles after they have logged in. There are three defined administrator roles: Network Administrator (NA), Security Administrator (SA), and Super Security Administrator (Super-SA). Each has specific responsibilities and, hence, access to corresponding functions and each must login through a console that is specifically assigned to their role.

Note that TOE is also designed to assign reasonable, restrictive, default values for the information flow policies.

The Security management function is designed to satisfy the following security functional requirements:

- FMT_MOF.1: The Super-SA modifies administrator user records. The SA modifies behavior of audit, MAC (sensitivity labels), content filtering and covert channel controls. The NA modifies behavior of IP DAC packet filters and routing.
- FMT_MSA.1: The SNS limits control of the sensitivity labels to the SA and SSA. This configuration is only allowed after successful login to the SA console.
- FMT_MSA.3: The SNS has default values for all security attributes. These defaults prohibit the network interfaces from accepting any traffic. The SA, SSA and NA are allowed to change the default values for some parameters.
- FMT_MTD.1: The SNS restricts the ability to create, delete, and modify administrator (SA and NA) accounts (user definitions) to the Super-SA.
- FMT_SMF.1: The SNS provides a comprehensive set of functions to manage its own security functions including, but not limited to: management of the Single-level Subscriber Information Flow Policy and Multi-level Subscriber Information Flow Policy including associated attributes, audit management and review, user attribute management, and Identification and Authentication configuration.
- FMT_SMR.2: Login to the appropriate console assigns the role. A user may only login to a console if the user is authorized for the role associated with that console. A user who successfully logs into the SA console is either a SA or Super-SA depending on the account type (role) associated with that user’s account attributes. The account type is set at account creation time by the Super-SA. Only the SA and Super-SA can log into the utility and SA consoles and the NA can log into only the NA console.

- FMT_SMR.3: The SNS requires a user to explicitly login to their corresponding role. At login, the console only displays a banner and a “login:” prompt. Each banner and prompt are the same amongst the UTY, SA and NA consoles. No help is provided to indicate valid user login names. The SA and Super-SA roles can only log into the utility console and SA console and the NA roles can only log into the NA console.

6.1.5 Protection of the TSF

The TOE has been designed to take advantage of the security features offered by the Intel x86 architecture, to isolate its own components to reduce the possibility of internal errors that might impact its security functions as well as to protect itself from users outside the TSF. The TOE offers only well defined subscriber device interfaces and administrator console interfaces that are designed to ensure that the applicable security functions are always invoked and succeed before allowing access to the services of the TOE.

The TOE has also been designed to resist and recover from a number of common failures (memory and disk access failures, network communication errors, power failures and unintentional resets). Since the TOE and its configuration are stored in non-volatile media, a power-cycle always restores the TOE to a secure state. The TOE is designed to test its own timer, memory, and PIC during start-up and to run additional tests periodically to check its own health. It is also designed to be able to limit throughput for its own protection, as well as to mitigate the potential for covert channels.

The Protection of the TSF function is designed to satisfy the following security functional requirements:

- FPT_AMT.1: The SNS runs confidence tests at start-up and periodically performs a memory test. The TOE is designed to test its own timer, memory, and PIC during start-up and to run memory tests periodically to check its own health.
- FPT_FLS.1: The SNS halts when it fails to successfully read or write from and to disk or memory. Given that the TOE configuration and security functions are non-volatile, a secure state is always restored on restart (e.g., after a power failure).
- FPT_RCV.3: The non-recoverable failures occur when the SNS loses power or is reset during database modification. In some cases, the database operation is partially completed. In those cases, the SNS deletes the affected records and identifies to the operator at the utility console (before SNS startup) which files have lost data. The operator can then restore the database from backup if a backup has been done, or add back the lost records using the command language. For resets and power outages where database updates are not in progress, the SNS returns to a secure state without user intervention. At most 3 database records may be lost during reset or power outage for the NM node. The SNS uses a transaction file to record the records being modified. On recovery, records in the transaction file are deleted as potentially inconsistent. The recovery state is consistent after these records are deleted. For all records or files deleted, the operator is informed on the utility console of the deletion. The operator is also informed of whether or not the file was able to be recovered.
- FPT_RCV.4: All security functions recover to a consistent and secure state after SNS power outage or reset since the TOE configuration and security functions are non-volatile and since the TOE has also been designed to resist and recover from a number of common failures (memory and disk access failures, network communication errors, power failures and unintentional resets).
- FPT_RVM.1: All inbound datagrams pass through the enforcement points for both the MAC and IP DAC packet filter policies. All outbound datagrams pass through the enforcement point for the MAC policy. Filtered proxy connections have filters placed in the communication path between the passive and active proxies, ensuring that the filter policy is enforced. If the datagram fails the content filter, the datagram is dropped. Each authorized user is required to be authenticated before they can perform any other function.
- FPT_SEP.3: The TSF is implemented in rings 0-2 of the Intel x86 architecture. The ring architecture protects the TSF from interference and tampering by least privilege processes. Process address space isolation is enforced by the ring 0 separation kernel. The TSF MAC and packet filter enforcement is performed by the IP process, and content filtering enforcement is performed by the generic filter (GF) process. These ring 2 processes can communicate with the rest of the TSF only through ring 0 separation kernel interfaces.

- FPT_STM.1: The SNS sets time at the Network Management node. All SNSs set their time off this clock and use that time for time-stamps to ensure consistency across a distributed configuration.
- FPT_TDC.1: Only CIPSO security labels are used between the SNS and another trusted IT product as security policy related data. The semantics of these are controlled by the CIPSO domain of interpretation. These labels are contained within IP headers. The SNS decodes the CIPSO labels per the standard to translate these labels to the SNS internal label format.
- FPT_TST.1: The SNS performs a self test on initial start-up to demonstrate correct operation of internal timer, memory, and PIC. The SNS integrity is verified at startup and the SNS maintains integrity or halts. The SNS code cannot be modified once the SNS is deployed. Note that there are no explicit operator-driven mechanisms to verify the integrity, though they can readily see the results/effects. The self test for the timer is performed on initial start-up and works by placing a numeric value in the countdown timer. The timer is instructed to fire an interrupt upon countdown to 0. Simultaneously, the test process is delayed slightly longer than the expected countdown. If the process wakes up before the countdown is finished and before the interrupt has fired, the TOE halts. The PIC test is also performed on initial start-up in a similar fashion, except the test watches for spurious interrupts during the countdown. The memory tests are performed continuously while the TOE is in operation. Test patterns are periodically written to and read back from memory and verified. Any memory error results in a system halt.
- FRU_FLT.2: The Network management node continues operation in spite of failure of remote non-NM SNSs or inter-SNS links. Remote SNSs continue to operate if they are configured for degraded mode operation when the Network Management node or inter-SNS link fails.
- FRU_RSA.1: The SNS enforces maximum quotas for datagram throughput over a specified period of time. NA controls subscriber device throughput constraints by setting the packets_per_second parameter in the *device* submenu of the NA console.

6.2 TOE Security Assurance Measures

6.2.1 Configuration management

Boeing uses automated configuration management tools to control and track changes to the applicable configuration items, to ensure that only authorized users can access or modify configuration items, to generate the TOE from the configuration items, and even to compare different releases to ensure that only the expected changes are included in a given release. A change review control board must review changes and has final authority over their acceptance into the TOE for a given release. The controlled set of configuration items include, but are not limited to: implementation representation, identified security flaws, development tools, and the evaluation evidence identified throughout this section of the Security Target (including the Security Target itself).

These activities are documented in:

- Boeing SNS Configuration Management Plan
- Boeing SNS Configuration Item List
- Boeing SNS FTLS Supporting Document List

The Configuration management assurance measure satisfies the following EAL 5 augmented with ACM_AUT.2, ACM_CAP.5, ADO_DEL.3, ADV_HLD.4, ADV_IMP.3, ADV_INT.3, ADV_LLD.2, ADV_RCR.3, ALC_DVS.2, ALC_FLR.2, ALC_LCD.3, ALC_TAT.3, ATE_COV.3, ATE_DPT.3, ATE_FUN.2, AVA_CCA.2, and AVA_MSU.3 assurance requirements:

- ACM_AUT.2
- ACM_CAP.5
- ACM_SCP.3

6.2.2 Delivery and operation

The TOE is delivered using a trusted commercial carrier, though a special trusted courier can be arranged. The TOE is a hardware appliance that is designed to check its own integrity at start-up. The TOE is delivered configured and ready to operate, but instructions are provided for connecting the TOE to its subscribers and associated administration consoles. These instructions also detail steps to ensure a secure installation, generation and startup of the TOE.

These activities are documented in:

- Boeing SNS Delivery Document
- Boeing SNS Operations and Maintenance Manual

The Delivery and operation assurance measure satisfies the following EAL 5 augmented with ACM_AUT.2, ACM_CAP.5, ADO_DEL.3, ADV_HLD.4, ADV_IMP.3, ADV_INT.3, ADV_LLD.2, ADV_RCR.3, ALC_DVS.2, ALC_FLR.2, ALC_LCD.3, ALC_TAT.3, ATE_COV.3, ATE_DPT.3, ATE_FUN.2, AVA_CCA.2, and AVA_MSU.3 assurance requirements:

- ADO_DEL.3
- ADO_IGS.1

6.2.3 Development

Boeing has numerous documents describing all facets of the design of the TOE. In particular, they have a functional specification that formally describes the accessible TOE interfaces; a high-level design that decomposes the TOE architecture into subsystems and formally describes each subsystem and their interfaces; a low-level design that further decomposes the TOE architecture into modules and semi-formally describes each module and their interfaces; and, correspondence documentation that explains (in appropriate formality) how each of the design abstractions corresponds from the TOE summary specification in the Security Target to the actual implementation of the TOE. Furthermore, Boeing has a formal security model that describes each of the security policies implemented by Boeing SNS. Of course, the implementation of the TOE itself is also available in its entirety and that implementation has been designed to be layered and otherwise structured to minimize its complexity.

These activities are documented in:

- Boeing SNS Formal Top-Level Specification
- Boeing SNS Detailed Top-Level Specification
- Boeing SNS High-level Design
- Boeing SNS Low-level Design
- Boeing SNS Security Policy Model
- Boeing SNS source code

The Development assurance measure satisfies the following EAL 5 augmented with ACM_AUT.2, ACM_CAP.5, ADO_DEL.3, ADV_HLD.4, ADV_IMP.3, ADV_INT.3, ADV_LLD.2, ADV_RCR.3, ALC_DVS.2, ALC_FLR.2, ALC_LCD.3, ALC_TAT.3, ATE_COV.3, ATE_DPT.3, ATE_FUN.2, AVA_CCA.2, and AVA_MSU.3 assurance requirements:

- ADV_FSP.3
- ADV_HLD.4
- ADV_IMP.3
- ADV_INT.3
- ADV_LLD.2
- ADV_RCR.3

- ADV_SPM.3

6.2.4 Guidance documents

Boeing provides administrator and user guidance on how to utilize the TOE security functions and warnings to administrators and users about actions that can compromise the security of the TOE.

These activities are documented in:

- Boeing SNS Trusted Facility Manual

The Guidance documents assurance measure satisfies the following EAL 5 augmented with ACM_AUT.2, ACM_CAP.5, ADO_DEL.3, ADV_HLD.4, ADV_IMP.3, ADV_INT.3, ADV_LLD.2, ADV_RCR.3, ALC_DVS.2, ALC_FLR.2, ALC_LCD.3, ALC_TAT.3, ATE_COV.3, ATE_DPT.3, ATE_FUN.2, AVA_CCA.2, and AVA_MSU.3 assurance requirements:

- AGD_ADM.1
- AGD_USR.1

6.2.5 Life cycle support

Boeing ensures the adequacy of the procedures used during the development and maintenance of the TOE through the use of a comprehensive life-cycle management plan. Boeing applies security controls on the development environment that are adequate to provide the confidentiality and integrity of the TOE design and implementation that is necessary to ensure the secure development of the TOE. Boeing has procedures that define the process for accepting and acting upon user reports of security flaws. These procedures describe the acceptance criteria for security flaws, how all security flaws and the status of fixes for each security flaw are tracked, and how corrections and corrective measures are made available as applicable. Boeing has a documented (standard and measurable) model of the TOE life cycle that ensures that the TOE is developed and maintained in a well-defined manner. Boeing uses well-defined development tools along with established implementation standards in order to ensure consistent and predictable results while developing the TOE.

These activities are documented in:

- Boeing SNS Life-cycle Model
- Boeing Development Environment Protection
- Boeing Configuration Maintenance Plan

The Life cycle support assurance measure satisfies the following EAL 5 augmented with ACM_AUT.2, ACM_CAP.5, ADO_DEL.3, ADV_HLD.4, ADV_IMP.3, ADV_INT.3, ADV_LLD.2, ADV_RCR.3, ALC_DVS.2, ALC_FLR.2, ALC_LCD.3, ALC_TAT.3, ATE_COV.3, ATE_DPT.3, ATE_FUN.2, AVA_CCA.2, and AVA_MSU.3 assurance requirements:

- ALC_DVS.2
- ALC_FLR.2
- ALC_LCD.3
- ALC_TAT.3

6.2.6 Tests

Boeing has a test plan that describes how each of the necessary security functions is tested, along with the expected test results. Boeing has documented each test as well as a rigorous analysis of test coverage and depth demonstrating that the security aspects of the design evident from the functional specification through the implementation representation are completely tested. Actual test results are created on a regular basis to demonstrate that the tests have been applied and that the TOE operates as designed.

These activities are documented in:

- Boeing SNS Test Plan
- Boeing SNS Test Procedures
- Being SNS Test Coverage and Depth Analysis
- Actual test results

The Tests assurance measure satisfies the following EAL 5 augmented with ACM_AUT.2, ACM_CAP.5, ADO_DEL.3, ADV_HLD.4, ADV_IMP.3, ADV_INT.3, ADV_LLD.2, ADV_RCR.3, ALC_DVS.2, ALC_FLR.2, ALC_LCD.3, ALC_TAT.3, ATE_COV.3, ATE_DPT.3, ATE_FUN.2, AVA_CCA.2, and AVA_MSU.3 assurance requirements:

- ATE_COV.3
- ATE_DPT.3
- ATE_FUN.2
- ATE_IND.2

6.2.7 Vulnerability assessment

The TOE administrator and user guidance documents describe the operation of Boeing SNS and how to maintain a secure state. These guides also describe all necessary operating assumptions and security requirements outside the scope of control of the TOE. They have been developed to serve as complete, clear, consistent, and reasonable administrator and user references. Furthermore, Boeing has conducted a misuse analysis demonstrating that the provided guidance is complete. Boeing has conducted a strength of function analysis wherein all permutational or probabilistic security mechanisms have been identified and analyzed resulting in a demonstration that all of the relevant mechanisms fulfill the minimum strength of function claim, SOF-high. Boeing performs regular vulnerability analyses of the entire TOE (including documentation) to identify weaknesses that can be exploited in the TOE. Boeing has also conducted covert channel analyses documenting a search for covert storage and timing channels that might be related to the information flow policies.

These activities are documented in:

- Boeing SNS Penetration Test Plan, Section 4, Vulnerability Analysis
- Boeing SNS Covert Channel Analysis

The Vulnerability assessment assurance measure satisfies the following EAL 5 augmented with ACM_AUT.2, ACM_CAP.5, ADO_DEL.3, ADV_HLD.4, ADV_IMP.3, ADV_INT.3, ADV_LLD.2, ADV_RCR.3, ALC_DVS.2, ALC_FLR.2, ALC_LCD.3, ALC_TAT.3, ATE_COV.3, ATE_DPT.3, ATE_FUN.2, AVA_CCA.2, and AVA_MSU.3 assurance requirements:

- AVA_CCA.2
- AVA_MSU.3
- AVA_SOF.1
- AVA_VLA.3

7. Protection Profile Claims

This Security Target makes no Protection Profile claims.

8. Rationale

This section provides the rationale for completeness and consistency of the Security Target. The rationale addresses the following areas:

- Security Objectives;
- Security Functional Requirements;
- Security Assurance Requirements;
- Strength of Functions;
- Requirement Dependencies;
- TOE Summary Specification; and,
- PP Claims.

8.1 Security Objectives Rationale

This section shows that all secure usage assumptions, organizational security policies, and threats are completely covered by security objectives. In addition, each objective counters or addresses at least one assumption, organizational security policy, or threat.

8.1.1 Security Objectives Rationale for the TOE and Environment

This section provides evidence demonstrating the coverage of organizational policies and usage assumptions by the security objectives.

	T.AUDIT	T.FILTER	T.I&A	T.MAC	T.OPERATE	A.ADMIN	A.COMMS	A.FLOW	A.PHYSEC	A.SUBSCRIBE
O.AUDLOS	X									
O.AUDREC	X									
O.AUDREV	X									
O.AUDTHR	X									
O.FILTER1		X								
O.FILTER2		X								
O.IDAUTH			X							
O.IMPEXP				X						
O.MAC1				X						
O.MAC2				X						
O.PROTECT					X					
O.RECOVER					X					
O.SELFTEST					X					
OE.ADMIN						X				
OE.COMMS							X			
OE.FLOW								X		
OE.PHYSEC									X	
OE.SUBSCRIBE										X

Table 3 Environment to Objective Correspondence

8.1.1.1 T.AUDIT

Attempts to violate TOE security policies may go undetected or users may not be accountable for security-relevant actions they perform.

This Threat is countered by ensuring that:

- O.AUDLOS: The TOE ensures that the TOE can limit the loss of audit information to prevent attempts to flood the audit trail in order to avoid accountability.
- O.AUDREC: The TOE ensures that an audit trail records security-related events with adequate contents (e.g., times and dates) so that actions are appropriately documented for accountability.
- O.AUDREV: The TOE ensures that the audit trail is protected so that only an administrator can effectively view or modify its contents preventing potential disclosure of sensitive information (e.g., user identities) and corruption of the accountability record.
- O.AUDTHR: The TOE ensures that administrators can establish thresholds to signal when security event thresholds have been exceeded so that evident attempts to violate a TOE security policy will be less likely to go unnoticed.

8.1.1.5 T.FILTER

Inappropriate network traffic may enter or leave a protected network.

This Threat is countered by ensuring that:

- O.FILTER1: The TOE ensures that information filtering rules can be defined by only an authorized administrator. It is important that an administrator can construct the rules to represent their own policies and that only an authorized administrator can do so to make sure the rules cannot be inappropriately changed (e.g., to make them ineffective).
- O.FILTER2: The TOE ensures that information can flow among subscriber devices only in accordance with filtering rules based on information headers and content. Obviously, rules are only effective if they are applied to actual information flows.

8.1.1.2 T.I&A

Unauthorized users may be able to inappropriately configure the TOE or access sensitive TOE data.

This Threat is countered by ensuring that:

- O.IDAUTH: The TOE ensures that administrators must be identified and authenticated before they can perform any other function. Allowing the functions to be accessed without proper authentication would allow the TOE policies to be arbitrarily configured resulting in ineffective security policies.

8.1.1.3 T.MAC

Classified information may be inappropriately accessed by entities that do not have appropriate clearances.

This Threat is countered by ensuring that:

- O.IMPEXP: The TOE ensures that labeled and unlabeled data is imported and exported in accordance with the sensitivity labels associated with attached subscriber devices. In order to control access appropriately, information must have appropriate security labels. In the case of multi-level devices, a security label within the device range must be associated with all information going to and coming from the device. In the case of single-level devices, only information at the level of the device must be sent to the device and all information coming from the device must be labeled with the device's security label.
- O.MAC1: The TOE ensures that subscriber device interfaces can be assigned security labels by only an authorized administrator.
- O.MAC2: The TOE ensures that information can flow between subscriber devices only if allowed based on the sensitivity labels of the associated subscriber devices and information being communicated. Whenever

information is sent or received, the security label of the information must be within the range of the security label(s) associated with the destination or source.

8.1.1.4 T.OPERATE

The TOE may fail to provide or enforce its security functions due to failure or malicious attacks against its security mechanisms.

This Threat is countered by ensuring that:

- O.PROTECT: The TOE is designed to ensure that its functions are not bypassable and are resistant to malicious attacks. While the TOE is primarily designed to enforce information flow policies, it also needs to protect itself such that its mechanisms cannot be bypassed or tampered with.
- O.RECOVER: The TOE provides the ability to resist and recover from common failure conditions. Failures might allow the TOE to continue to operate while its security policies may no longer be enforced. Hence, detection and recovery from the most common failures serves to mitigate the risk of inappropriate information flows during those circumstances.
- O.SELFTEST: The TOE provides self-testing functions to determine that it is operating correctly. By testing itself periodically, the TOE can determine that it seems to be working properly and its security policies are still being enforced.

8.1.1.6 A.ADMIN

The TOE administrators are competent, adhere to the applicable guidance, and are not willfully negligent or malicious.

This Assumption is satisfied by ensuring that:

- OE.ADMIN: This objective directly addresses the corresponding assumption.

8.1.1.7 A.COMMS

The TOE is able to communicate appropriately with its attached subscriber devices.

This Assumption is satisfied by ensuring that:

- OE.COMMS: This objective directly addresses the corresponding assumption.

8.1.1.8 A.FLOW

Protected information does not flow among the network subscribers unless it passes through the TOE.

This Assumption is satisfied by ensuring that:

- OE.FLOW: This objective directly addresses the corresponding assumption.

8.1.1.9 A.PHYSEC

The TOE is physically secure; specifically it, including the communication media among distributed parts of the TOE, is protected from physical tampering of itself or its physical connections to its environment (subscriber devices).

This Assumption is satisfied by ensuring that:

- OE.PHYSEC: This objective directly addresses the corresponding assumption.

8.1.1.10 A.SUBSCRIBE

A process outside the scope or control of the TOE is used to determine the attributes (e.g., sensitivity ranges) of attached subscriber devices.

This Assumption is satisfied by ensuring that:

- OE.SUBSCRIBE: This objective directly addresses the corresponding assumption.

8.2 Security Requirements Rationale

This section provides evidence supporting the internal consistency and completeness of the components (requirements) in the Security Target. Note that **Table 4** indicates the requirements that effectively satisfy the individual objectives..

8.2.1 Security Functional Requirements Rationale

All Security Functional Requirements (SFR) that are identified in this Security Target are fully addressed in this section and each SFR is mapped to the objective that it is intended to satisfy.

	O.AUDLOS	O.AUDREC	O.AUDREV	O.AUDTHR	O.FILTER1	O.FILTER2	O.IDAUTH	O.IMPEXP	O.MAC1	O.MAC2	O.PROTECT	O.RECOVER	O.SELFTEST
FAU_ARP.1				X									
FAU_GEN.1		X											
FAU_SAA.1				X									
FAU_SAR.1			X										
FAU_SAR.2			X										
FAU_SEL.1		X											
FAU_STG.2	X												
FAU_STG.3	X												
FAU_STG.4	X												
FDP_ETC.1								X					
FDP_ETC.2								X					
FDP_IFC.2						X				X			
FDP_IFF.2						X				X			
FDP_IFF.4										X			
FDP_ITC.1								X					
FDP_ITC.2								X					
FDP_RIP.2										X			
FIA_AFL.1							X						
FIA_ATD.1							X						
FIA_SOS.1							X						
FIA_UAU.1							X						
FIA_UAU.7							X						
FIA_UID.2							X						
FMT_MOF.1					X		X		X				
FMT_MSA.1									X				
FMT_MSA.3						X				X			
FMT_MTD.1													
FMT_SAE.1							X						
FMT_SMF.1					X		X						
FMT_SMR.2							X						
FMT_SMR.3							X						
FPT_AMT.1													X
FPT_FLS.1												X	
FPT_RCV.3												X	

FPT_RCV.4												X	
FPT_RVM.1											X		
FPT_SEP.3											X		
FPT_STM.1		X											
FPT_TDC.1										X			
FPT_TST.1													X
FRU_FLT.2												X	
FRU_RSA.1											X		
FTA_TAH.1							X						

Table 4 Objective to Requirement Correspondence

8.2.1.1 O.AUDLOS

The TSF shall be configurable to limit the potential loss of audit information.

This TOE Security Objective is satisfied by ensuring that:

- FAU_STG.2: The TOE prevents unauthorized modification to audit records and ensures that none will be lost if the audit storage space is exhausted. Protecting access to audit records and requiring preservation of events when the audit trail becomes full serves to limit loss of audit data.
- FAU_STG.3: The TOE will warn the SA/SSA if the audit trail exceeds 90% of its capacity to mitigate the chance of audit storage space exhaustion. Warning the administrator about the imminent exhaustion of the available audit space allows audit records to be archived and audit storage space to be recovered.
- FAU_STG.4: The TOE can be configured by the SA/SSA to either prevent auditable events or overwrite old audit events when the audit trail becomes full. The administrator can choose whether it is more important to prevent the loss of audit records or to preserve system functions at the cost of losing audit records.

8.2.1.2 O.AUDREC

The TOE shall provide a means to record an audit trail of security-related events, with accurate dates and times.

This TOE Security Objective is satisfied by ensuring that:

- FAU_GEN.1: The TOE records a log of security relevant events, including date, time, event type, subject, and outcome, as they occur. Obviously, if audit records are not generated, there is no accountability at all.
- FAU_SEL.1: The TOE provides the ability to limit the generated audit records based on event type and sensitivity level in order to help limit any audit records that might be deemed unnecessary. This requirement allows the administrator to decide which events they want to actually audit so that unwanted audit records can be avoided to save space and perhaps to ease audit review.
- FPT_STM.1: The TOE generates a time stamp for use in audit records. Without time stamps, audit records cannot be sequenced or do not serve to indicate when particular events occurred.

8.2.1.3 O.AUDREV

The TSF shall protect the audit trail so that only an authorized administrator can access the audit trail.

This TOE Security Objective is satisfied by ensuring that:

- FAU_SAR.1: The TOE provides the SA/SSA the ability to review all available audit data. Obviously without the ability to review audit data the entire audit function is essentially pointless.
- FAU_SAR.2: The TOE allows only authorized administrators to access audit data. Audit records may contain sensitive information that should not be generally available.

8.2.1.4 O.AUDTHR

The TSF shall allow audit thresholds to be defined that will trigger alarms when attempted policy violations exceed the defined thresholds.

This TOE Security Objective is satisfied by ensuring that:

- FAU_ARP.1: The TOE will send a warning to the administrator console when potential security violations are detected. Some audit events are more important than others and when particularly important security events occur, it is important to bring that more directly and immediately to the administrator's attention.
- FAU_SAA.1: The TOE monitors audit events in order to detect potential security policy violations. While some audit events may be more important than others, there are cases where the accumulation of events may indicate something that a single event does not. Hence, it is important to be able to identify such events based on accumulation statistics.

8.2.1.5 O.FILTER1

The TOE shall allow (only) an authorized administrator to explicitly define information filtering rules.

This TOE Security Objective is satisfied by ensuring that:

- FMT_MOF.1: The TOE limits the ability to modify the behavior of the information flow policies to an authorized administrator. If non-administrators were able to configure the information flow rules, the information flow policy would not be particularly effective.
- FMT_SMF.1: The TOE provides the ability to manage the information flow policies. If the information flow rules cannot be configured, then the TOE cannot be tailored to the specific circumstances of its environment.

8.2.1.6 O.FILTER2

The TOE shall restrict the flow of information among subscriber devices based on filtering rules based on information headers and content established by the authorized administrator.

This TOE Security Objective is satisfied by ensuring that:

- FDP_IFC.2: The TOE enforces its information flow policies on all information flowing among subscriber devices. If the information flow policy did not apply to all applicable subjects, objects, and information then the TOE would not be effective in controlling the flow of information among its subscribers.
- FDP_IFF.2: The TOE enforces filter based information flow rules when processing attempted information flows among subscriber devices. If the information flow rules are not well defined and consistently applied, the information flow policy may not be particularly effective.
- FMT_MSA.3: The TOE ensures that the information flow policies are restrictive by default. By enforcing restrictive defaults, the TOE will ensure that violations to the intended policy would not occur prior to establishing the proper rules.

8.2.1.7 O.IDAUTH

The TOE shall uniquely identify and authenticate the claimed identity of all administrators before granting access to TOE functions related to the assumed administrator role.

This TOE Security Objective is satisfied by ensuring that:

- FIA_AFL.1: The TOE detects failed attempts at user authentication and imposes a delay in order to mitigate attempts to bypass the authentication mechanism by guessing passwords. By hampering attempts to guess the login credentials of others, the TOE helps to mitigate the possibility of successful improper authentication attempts.
- FIA_ATD.1: The TOE maintains user identities, roles, and authentication data in support of identification and authentication, including assumption of a specific role. Without well-defined attributes, the TOE would be unable to identify and authenticate users and assign appropriate credentials once authenticated.
- FIA_SOS.1: The TOE enforces password composition rules to help ensure that passwords would be difficult to guess. By ensuring that passwords have at least some minimal measure of complexity, the TOE helps to mitigate the possibility of successful improper authentication attempts.

- FIA_UAU.1: The TOE requires administrators to be authenticated prior to accessing TOE functions. By requiring users to authenticate themselves, the TOE can have assurance that the identified user is actually the right user.
- FIA_UAU.7: The TOE ensures that only obscured feedback is provided when entering authentication data to mitigate the possibility of inappropriately disclosing that data. This mitigates the possibility of a user observing the authentication credentials of another user which might allow improper authentication.
- FIA_UID.2: The TOE requires every user (administrators and subscriber devices) to be identified before accessing any functions.
- FMT_MOF.1: The TOE restricts the ability to manage the identification and authentication function to authorized administrators. The identification and authentication function is dependent on the ability to define valid user accounts and those accounts should only be definable by appropriately authorized administrators, otherwise the mechanism could be subverted.
- FMT_SAE.1: The TOE enforces user account and password expirations that can be configured by the Super-SA. By requiring passwords to be changed on a regular basis, the TOE mitigates the possibility of long-term password guessing .
- FMT_SMF.1: The TOE provides the ability to manage the user account definitions. The identification and authentication function is dependent on the ability to define valid user accounts.
- FMT_SMR.2: The TOE provides a set of roles associated with authorized administrators upon successful logon. The TOE assigned specific security functions to specific roles and those roles to specific users so as to share the security responsibilities.
- FMT_SMR.3: The TOE requires that each administrator must intentionally assume one of the available administrator roles when logging on. The explicit assumption of administrator roles serves to provide those users the least or specific privileges necessary to perform their assigned functions.
- FTA_TAH.1: The TOE informs the user of the date and time of their last successful and unsuccessful logon attempt when successfully logging on in order to mitigate the possibility of undetected attempts to guess a password. By informing the user of login attempts that they may not have made, the users can become aware of attempts to login improperly.

8.2.1.8 O.IMPEXP

The TOE shall import and export labeled and unlabelled data according to the sensitivity labels associated with attached subscriber devices.

This TOE Security Objective is satisfied by ensuring that:

- FDP_ETC.1: The TOE sends unlabelled information to single-level subscriber devices. Single-level devices are not aware of labels and any information sent to those devices should be at the right level, but should not be labeled.
- FDP_ETC.2: The TOE sends labeled information to multi-level subscriber devices. Multi-level devices are trusted to properly handle labeled information within their defined range; as such they need to be aware of the specific security labels of the information they receive.
- FDP_ITC.1: The TOE accepts unlabelled information from single-level subscriber devices and labels it according to the label of the subscriber device. Given that single-level devices are not aware of labels, it is necessary to label any information received from those devices with the label of the device itself so it can be appropriately controlled.
- FDP_ITC.2: The TOE accepts labeled information from multi-level subscriber devices. Given that multi-level devices are trusted to properly handle labeled information, information received from those devices should have labels when received (and those labels should be within the range of the device).

8.2.1.9 O.MAC1

The TOE shall allow (only) an authorized administrator to assign sensitivity labels to subscriber devices.

This TOE Security Objective is satisfied by ensuring that:

- FMT_MOF.1: The TOE limits the ability to modify the behavior of the information flow policies to an authorized administrator. If the mandatory information flow rules could be changed by a non-administrator, the information flow policy could become ineffective.

- FMT_MSA.1: The TOE limits the ability to modify sensitivity labels to the SA/SSA. Security labels serve a primary role in controlling how information is labeled and where it can flow and as such these labels must be strictly controlled.
- FMT_SMF.1: The TOE provides the ability to modify sensitivity labels. Just as security label changes must be appropriately restricted, they must also be assignable to effectively manage and define the mandatory information flow policy.

8.2.1.10 O.MAC2

The TOE shall restrict the flow of information between attached subscriber devices so that information from one subscriber can be sent to another subscriber only if the sensitivity level of the information is within the range of sensitivity labels the receiving subscriber device is allowed to process.

This TOE Security Objective is satisfied by ensuring that:

- FDP_IFC.2: The TOE enforces its information flow policies on all information flowing among subscriber devices. If the information flow policy did not apply to all applicable subjects, objects, and information then the TOE would not be effective in controlling the flow of information among its subscribers.
- FDP_IFF.2: The TOE enforces appropriate sensitivity-label based information flow rules when processing attempted information flows among subscriber devices. If the information flow rules are not well defined and consistently applied, the information flow policy may not be effective.
- FDP_IFF.4: The TOE limits the ability to send information covertly among subscriber devices using covert timing or storage channels. Limiting the use of covert channels helps to mitigate the risk of bypassing the information flow rules.
- FDP_RIP.2: The TOE ensures that information is not allowed to flow inappropriately due to a reuse of previously processed information. Ensuring that residual information is not allowed to be accessed helps mitigate the risk of bypassing the information flow rules.
- FMT_MSA.3: The TOE ensures that the information flow policies are restrictive by default. By enforcing restrictive defaults, the TOE will ensure that violations to the intended policy would not occur prior to establishing the proper rules.
- FPT_TDC.1: The TOE ensures that sensitivity labels are interpreted in accord with standards. If security labels are not interpreted in standard fashion, then inconsistencies in the policy could arise allowing inappropriate information to flow among the intended subjects.

8.2.1.11 O.PROTECT

The TOE shall ensure that its functions are always invoked and that it is resistant to potential attacks against its security functions.

This TOE Security Objective is satisfied by ensuring that:

- FPT_RVM.1: The TOE ensures that its own security functions are not bypassable. If the security policies could be bypassed, then they are not enforced.
- FPT_SEP.3: The TOE is designed to separate its subjects from one another and from itself; it is further developed to be internally organized to protect its own security functions from one another. By protecting itself and being able to distinguish the security properties of its subjects, the TOE can protect itself and ensure its security policies can be enforced.
- FRU_RSA.1: The TOE is able to limit datagram throughput through an interface, to balance quality of service or to mitigate potential covert channels.

8.2.1.12 O.RECOVER

The TOE shall remain secure and be able to recover from failure conditions and will continue to operate when possible.

This TOE Security Objective is satisfied by ensuring that:

- FPT_FLS.1: The TOE ensures that it remains in a secure state when power fails and disk and memory errors occur. When disk and memory failures occur, the TOE resets itself and when it restarts (including after a power failure) the TOE configuration is restored to its non-volatile configuration.

- FPT_RCV.3: The TOE enters a recovery mode when automated recovery is not possible after a power failure or unexpected reset. Basically, if the TOE detects a situation from which it cannot recover, it puts itself into a state requiring administrator intervention to avoid any possibility of operating in a corrupted or otherwise inappropriate state.
- FPT_RCV.4: The TOE is able to recover a secure state after a power failure or unexpected reset. Given that the TOE configuration is non-volatile, a reset will result in bringing the TOE into its initial secure state.
- FRU_FLT.2: The Network management node continues operation in spite of failure of remote non-NM SNSs or inter-SNS link. Remote SNSs continue to operate if they are configured for degraded mode operation when the Network Management node or inter-SNS link fails.

8.2.1.13 O.SELFTEST

The TOE shall test its own operation in order to detect potential failures.

This TOE Security Objective is satisfied by ensuring that:

- FPT_AMT.1: The TOE tests its hardware components during start-up and periodically during normal operation to ensure they are working properly. The TOE is designed to test its underlying hardware and to halt on hardware failure to avoid the possibility of operating in an insecure state.
- FPT_TST.1: The TOE tests itself during start-up to ensure the correct operation of its interval timer, memory, and PIC. The TOE is designed to test aspects of its own operation so that it can halt if errors are detected to mitigate the possibility of continuing to operate in a potentially insecure state.

8.3 Security Assurance Requirements Rationale

Boeing Secure Network Server (SNS) has historically been evaluated using the Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria A1 level. This level has been compared with Common Criteria EAL 7. Accordingly, the SNS development process was designed to fulfill the requirements for EAL 7 augmented with ALC_FLR.2. This target departs from that template in four places: ADV_FSP, ADV_HLD, ATE_IND, and AVA_VLA.

All elements of the SNS kernel and infrastructure, including that for IP routing and MLS policy, were modeled and evaluated formally. In the original development of evidence for EAL7, the CDS (content filter) application interfaces, by virtue of the fact that they do not cross privilege boundaries, were treated as internal interfaces and specified semiformally. With the partitioning of some portions of the evaluation into infrastructure and CDS application components, however, the CDS interfaces could be considered external and, as such, require formal specification of their internals to meet ADV_FSP.4 and ADV_HLD.5 for EAL 7. Their representation therefore meets only ADV_FSP.3 and ADV_HLD.4 semiformal requirements. Similarly, the partitioning of the evaluation leads to the requirement under ATE_IND.3 that the evaluators independently develop, build and test applications within the programming environment. This was not done, due to resource limitations and legacy elements in the SNS development environment.

The Boeing Penetration Test Plan was written against and cites the requirements of AVA_VLA.4, and so includes justification of complete TOE coverage, per Content and Presentation of Evidence Element AVA_VLA.4.6C. The evaluator analyzed the system per AVA_VLA.3. For this reason, the SNS meets only AVA_VLA.3.

Therefore, the SNS is designated as EAL 5 augmented with ACM_AUT.2, ACM_CAP.5, ADO_DEL.3, ADV_HLD.4, ADV_IMP.3, ADV_INT.3, ADV_LLD.2, ADV_RCR.3, ALC_DVS.2, ALC_FLR.2, ALC_LCD.3, ALC_TAT.3, ATE_COV.3, ATE_DPT.3, ATE_FUN.2, AVA_CCA.2, and AVA_MSU.3.

While the TOE is expected to be physically secure and securely connected to its subscriber [SAR1] devices, no assumptions are made about the nature of those subscriber devices. The TOE has been designed to withstand any attempts that might be made to subvert its security policies.

8.4 Strength of Functions Rationale

The claim of SOF-high is applicable only to identification and authentication (FIA_UAU.1), the only Security Functional Requirement of a probabilistic or permutational nature.

8.5 Requirement Dependency Rationale

As can be seen in the table below, most of the dependencies as defined in the Common Criteria Parts 2 and 3 have been satisfied. The requirements identified in green text indicate areas where the dependencies are exceeded and requirements that are underlined represent dependencies satisfied with assurance requirements.

Notice that one dependency for FTP_ITC.1 remains unfulfilled. The dependency in effect requires a trusted channel or path between the TOE and its subscriber devices. The CIPSO labels related to FTP_ITC.2 are carried in the datagram headers and do not require any trusted channel for delivery given that the physical link to the subscriber is assumed to be protected.

ST Requirement	CC Dependencies	ST Dependencies
FAU_ARP.1	FAU_SAA.1	FAU_SAA.1
FAU_GEN.1	FPT_STM.1	FPT_STM.1
FAU_SAA.1	FAU_GEN.1	FAU_GEN.1
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1
FAU_SAR.2	FAU_SAR.1	FAU_SAR.1
FAU_SEL.1	FAU_GEN.1 and FMT_MTD.1	FAU_GEN.1 and FMT_MTD.1
FAU_STG.2	FAU_GEN.1	FAU_GEN.1
FAU_STG.3	FAU_STG.1	FAU_STG.2
FAU_STG.4	FAU_STG.1	FAU_STG.2
FDP_ETC.1	(FDP_ACC.1 or FDP_IFC.1)	FDP_IFC.2
FDP_ETC.2	(FDP_ACC.1 or FDP_IFC.1)	FDP_IFC.2
FDP_IFC.2	FDP_IFF.1	FDP_IFF.2
FDP_IFF.2	FDP_IFC.1 and FMT_MSA.3	FDP_IFC.2 and FMT_MSA.3
FDP_IFF.4	AVA_CCA.1 and FDP_IFC.1	AVA_CCA.2 and FDP_IFC.2
FDP_ITC.1	(FDP_ACC.1 or FDP_IFC.1) and FMT_MSA.3	FDP_IFC.2 and FMT_MSA.3
FDP_ITC.2	(FDP_ACC.1 or FDP_IFC.1) and (FTP_TRP.1 or FTP_ITC.1) and FPT_TDC.1	FDP_IFC.2 and [FTP_ITC.1] and FPT_TDC.1
FDP_RIP.2	none	none
FIA_AFL.1	FIA_UAU.1	FIA_UAU.1
FIA_ATD.1	none	none
FIA_SOS.1	none	none
FIA_UAU.1	FIA_UID.1	FIA_UID.2
FIA_UAU.7	FIA_UAU.1	FIA_UAU.1
FIA_UID.2	none	none
FMT_MOF.1	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.2 and FMT_SMF.1
FMT_MSA.1	FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	FMT_SMR.2 and FMT_SMF.1 and FDP_IFC.2
FMT_MSA.3	FMT_MSA.1 and FMT_SMR.1	FMT_MSA.1 and FMT_SMR.2
FMT_MTD.1	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.2 and FMT_SMF.1
FMT_SAE.1	FMT_SMR.1 and FPT_STM.1	FMT_SMR.2 and FPT_STM.1
FMT_SMF.1	none	none
FMT_SMR.2	FIA_UID.1	FIA_UID.2
FMT_SMR.3	FMT_SMR.1	FMT_SMR.2
FPT_AMT.1	none	none
FPT_FLS.1	none	none

FPT_RCV.3	FPT_TST.1 and AGD_ADM.1 and ADV_SPM.1	FPT_TST.1 and AGD_ADM.1 and ADV_SPM.3
FPT_RCV.4	ADV_SPM.1	ADV_SPM.3
FPT_RVM.1	none	none
FPT_SEP.3	none	none
FPT_STM.1	none	none
FPT_TDC.1	none	none
FPT_TST.1	FPT_AMT.1	FPT_AMT.1
FRU_FLT.2	FPT_FLS.1	FPT_FLS.1
FRU_RSA.1	none	none
FTA_TAH.1	none	none
ACM_AUT.2	ACM_CAP.3	ACM_CAP.5
ACM_CAP.5	ALC_DVS.2	ALC_DVS.2
ACM_SCP.3	ACM_CAP.3	ACM_CAP.5
ADO_DEL.3	ACM_CAP.3	ACM_CAP.5
ADO_IGS.1	AGD_ADM.1	AGD_ADM.1
ADV_FSP.3	ADV_RCR.1	ADV_RCR.3
ADV_HLD.4	ADV_FSP.3 and ADV_RCR.2	ADV_FSP.3 and ADV_RCR.3
ADV_IMP.3	ADV_INT.1 and ADV_LLD.1 and ADV_RCR.1 and ALC_TAT.1	ADV_INT.3 and ADV_LLD.2 and ADV_RCR.3 and ALC_TAT.3
ADV_INT.3	ADV_IMP.2 and ADV_LLD.1	ADV_IMP.3 and ADV_LLD.2
ADV_LLD.2	ADV_HLD.3 and ADV_RCR.2	ADV_HLD.4 and ADV_RCR.3
ADV_RCR.3	none	none
ADV_SPM.3	ADV_FSP.1	ADV_FSP.3
AGD_ADM.1	ADV_FSP.1	ADV_FSP.3
AGD_USR.1	ADV_FSP.1	ADV_FSP.3
ALC_DVS.2	none	none
ALC_FLR.2	none	none
ALC_LCD.3	none	none
ALC_TAT.3	ADV_IMP.1	ADV_IMP.3
ATE_COV.3	ADV_FSP.1 and ATE_FUN.1	ADV_FSP.3 and ATE_FUN.2
ATE_DPT.3	ADV_HLD.2 and ADV_IMP.2 and ADV_LLD.1 and ATE_FUN.1	ADV_HLD.4 and ADV_IMP.3 and ADV_LLD.2 and ATE_FUN.2
ATE_FUN.2	none	none
ATE_IND.2	ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1 and ATE_FUN.1	ADV_FSP.3 and AGD_ADM.1 and AGD_USR.1 and ATE_FUN.2
AVA_CCA.2	ADV_FSP.2 and ADV_IMP.2 and AGD_ADM.1 and AGD_USR.1	ADV_FSP.3 and ADV_IMP.3 and AGD_ADM.1 and AGD_USR.1
AVA_MSU.3	ADO_IGS.1 and ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1	ADO_IGS.1 and ADV_FSP.3 and AGD_ADM.1 and AGD_USR.1
AVA_SOF.1	ADV_FSP.1 and ADV_HLD.1	ADV_FSP.3 and ADV_HLD.4
AVA_VLA.3	ADV_FSP.1 and ADV_HLD.2 and ADV_IMP.1 and ADV_LLD.1 and AGD_ADM.1 and AGD_USR.1	ADV_FSP.3 and ADV_HLD.4 and ADV_IMP.3 and ADV_LLD.2 and AGD_ADM.1 and AGD_USR.1

8.6 Explicitly Stated Requirements Rationale

There are no explicitly defined requirements defined in this security target.

8.7 TOE Summary Specification Rationale

Each subsection in Section 6, the TOE Summary Specification, describes a security function of the TOE. Each description is followed with rationale that indicates which requirements are satisfied by aspects of the corresponding security function. The set of security functions work together to satisfy all of the security functions and assurance requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality.

This Section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TOE summary specification are all necessary for the required security functionality in the TSF. **Table 5 Security Functions vs. Requirements Mapping** demonstrates the relationship between security requirements and security functions.

	Security audit	**User data protection**	**Identification and authentication**	**Security management**	**Protection of the TSF**
FAU_ARP.1	X				
FAU_GEN.1	X				
FAU_SAA.1	X				
FAU_SAR.1	X				
FAU_SAR.2	X				
FAU_SEL.1	X				
FAU_STG.2	X				
FAU_STG.3	X				
FAU_STG.4	X				
FDP_ETC.1		X			
FDP_ETC.2		X			
FDP_IFC.2		X			
FDP_IFF.2		X			
FDP_IFF.4		X			
FDP_ITC.1		X			
FDP_ITC.2		X			
FDP_RIP.2		X			
FIA_AFL.1			X		
FIA_ATD.1			X		
FIA_SOS.1			X		
FIA_UAU.1			X		
FIA_UAU.7			X		
FIA_UID.2			X		
FMT_MOF.1				X	
FMT_MSA.1				X	
FMT_MSA.3				X	

FMT_MTD.1				X	
FMT_SAE.1			X		
FMT_SMF.1				X	
FMT_SMR.2				X	
FMT_SMR.3				X	
FPT_AMT.1					X
FPT_FLS.1					X
FPT_RCV.3					X
FPT_RCV.4					X
FPT_RVM.1					X
FPT_SEP.3					X
FPT_STM.1					X
FPT_TDC.1					X
FPT_TST.1					X
FRU_FLT.2					X
FRU_RSA.1					X
FTA_TAH.1			X		

Table 5 Security Functions vs. Requirements Mapping

8.8 PP Claims Rationale

See Section 7, Protection Profile Claims.