XTS-400 UK EAL5 Security Target - XTS-400 Version 6.4(UKE)

November 2007 – Version 02
SAP Document No: A2N 45009531
EPF Reference: 2.1.4

BAE SYSTEMS

“Linux” is a registered trademark of Linus Torvalds.

“Red Hat” and “RPM” (Red Hat Package Manager) are registered trademarks of Red Hat Software, Inc. in the United States and other countries.

“STOP”, “SAGE”, “XTS-300” and “XTS-400” are trademarks of BAE Systems Information Technology, LLC.

“Intel” and “Pentium” are registered trademarks of the Intel corp. “Xeon” is a trademark of the Intel Corp.

"Netscape” is a registered trademark of Netscape Communications Corporation in the U.S. and other countries.

“UNIX” is a registered trademark of The Open Group.

“Apache” is a trademark of the Apache Software Foundation.

All other product names mentioned herein are trademarks of their respective owners.

The TCP/IP software contained in this release is derived from material which is copyright Regents of the University of California. The following paragraph applies to that software in its original form as provided by U.C.:

> “1. Redistribution of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistribution in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. All advertising materials mentioning features or use of this software must display the following acknowledgment: ‘This product includes software developed by the University of California, Berkeley and its contributors.’ 4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.”

Various STOP application programs may use or link to one or more libraries, including uClibc, which are licensed under Version 2.1 of the GNU Lesser General Public License

("LGPL") available at `http://www.gnu.org/copyleft/lesser.html`. In accordance with the LGPL, the source code for all linked libraries is provided free of charge on the Applications CD. BAE Systems Information Technology, LLC claims no interest or ownership, including copyrights, in any of the linked libraries licensed under the LGPL.

List of Contents

List of Contents ..... *iv*
List of Tables ..... *vi*
Document Distribution ..... *vii*
Document History ..... *ix*
List of Abbreviations ..... *xi*

Section 1 Introduction 1
1.1 Security Target Identification 1
1.2 Security Target Overview 1
1.2.1 XTS-400 Background and History 2
1.3 CC Conformance Claim 3
1.4 Conventions 3
1.5 Terminology 4
Section 2 TOE Description 9
2.1 Product Description 9
2.1.1 Software Overview 9
2.1.2 Hardware Overview 10
2.1.3 Minimal Evaluated Configuration 10
2.1.4 TOE Definition 13
2.2 General TOE Functionality 14
2.2.1 Security Features 14
2.2.2 Other Characteristics of the TOE 14
Section 3 TOE Security Environment 17
3.1 Secure Usage Assumptions 17
3.2 Security Threats 18
3.3 Organisational Security Policies 23
Section 4 Security Objectives 25
4.1 Security Objectives for the TOE 25
4.2 Security Objectives for the Environment 26
Section 5 IT Security Requirements 29
5.1 TOE Security Functional Requirements 31
5.1.1 5.1.1 Security audit (FAU) 31
5.1.2 User data protection (FDP) 34
5.1.3 Identification and authentication (FIA) 44
5.1.4 Security management (FMT) 48
5.1.5 Protection of the TOE Security Functions (FPT) 56
5.1.6 TOE Access (FTA) 57
5.1.7 Trusted path/channels (FTP) 59
5.2 End Notes 59

5.3 TOE Security Assurance Requirements....66
5.4 Strength of TOE Security Functional Requirements....67
5.5 Security Requirements for the IT Environment....67
Section 6 TOE Summary Specification....69
6.1 Measures Used to Meet IT Security Functions....69
6.1.1 Audit Generation – AUDGEN....76
6.1.2 Audit Review – AUDREV....77
6.1.3 Discretionary Access Control – DACENF....77
6.1.4 Mandatory Access Control – MACENF....78
6.1.5 Mandatory Integrity Control – MICENF....78
6.1.6 Residual Information Protection – RIPENF....79
6.1.7 Identification and Authentication – IDNAUT....79
6.1.8 Security Management – SECMGT....80
6.1.9 Security Function Protection – SECFPR....81
6.1.10 Trusted Path – TRUPTH....86
6.2 Assurance Measures....86
6.3 IT Security Functions Realised by Probabilistic or Permutational Mechanisms....89
Section 7 PP Claims....91
7.1 PP Reference....91
7.2 PP Refinements, Selections, and Assignments....91
7.3 PP Additions....91
Section 8 Rationale....95
8.1 Security Objectives Rationale....95
8.2 Security Requirements Rationale....104
8.2.1 Functional Security Requirements Rationale....104
8.2.2 Assurance Security Requirements Rationale....108
8.2.3 Rationale that IT Security Requirements are Internally Consistent....108
8.3 Functional Requirements Grounding in Objectives....109
8.4 PP Claims Rationale....114
8.5 Strength-of-Function Rationale....115
8.6 Rationale for Inclusion of Hardware in the TOE....116
Section 9 Appendix A - List of Subjects....119
9.1 Subjects....119
9.1.1 Subject Attributes....119
9.1.2 Subject Creation and Destruction....119
Section 10 Appendix B – List of Objects....121
10.1 Information and Objects....121
10.1.1 Semaphores....121
10.1.2 Processes....121
10.1.3 Devices....122
10.1.4 Sockets....122
10.1.5 File System Objects....123

10.1.6 Memory Objects 125

Section 11 Appendix C – TSF Commands 127

Section 12 Appendix D – User Commands 131

Section 13 Appendix E – Auditable Events 135

Section 14 Appendix F – Selectable Audit Events 139

Section 15 Appendix G -Acronyms 141

Section 16 Appendix H – Functional Requirements Required by Existing XTS-400 Customers 143

List of Tables

Table 1: Security Threats 18
Table 2: Security Functional Requirements 29
Table 3: Assurance Requirements – EAL5 Augmented 66
Table 4: Security Functions and the SFRs They Implement 69
Table 5: SFRs and the Security Functions that Satisfy Them 70
Table 6: Security Assurance Requirements and the Security Measures That Meet Them 87
Table 7: Mapping the TOE Security Environment to Security Objectives 95
Table 8: Tracing of Security Objectives to the TOE Security Environment 101
Table 9: Functional and Assurance Component to Security Objective Mapping 104
Table 10: Requirements to Objectives Mapping 109
Table 11: LSPP Objectives to ST Objectives Mapping 114

Document Distribution

Company Name	Addressee's Name(s)	No. of Copies
CESG	Certifier Deputy Certifier	1 (PDF format) 1 (PDF format)
LogicaCMG	Dr Steve Hill	1 (PDF format)
BAE Systems	Author EPF	1 (PDF & .doc) 1 (PDF & .doc)

This page is intentionally left blank

Document History

Version	Date	Details of Change
01	July 2007	ADC No. - XTS/002, Change No. - 397359, Initial issue.
02	November 2007	ADC No. – XTS/003, Change No. - 402091, Updated in response to CESG & CLEF comments.

X

This page is intentionally left blank

List of Abbreviations

AD	Application Domain
COTS	Commercial off the shelf
DAC	Discretionary Access Control
EAL	Evaluation Assurance Levels
EPF	Electronic Project Files
MAC	Mandatory Access Control
MIC	Mandatory Integrity Control
OSI	Open Systems Interconnection
OSS	Operating System Services
SAK	Secure Attention Key
ST	Security Target
STOP	Secure Trusted Operating Program
TFM	Trusted Facilities Manual
TOE	Target of Evaluation
TSF	TOE Security Functions
TSS	Trusted System Services

This page is intentionally left blank

Section 1 Introduction

1.1 Security Target Identification

Title: Security Target for XTS-400, Version 6.4(UKE)

Keywords: MAC, DAC, MIC, Mandatory Access Control, Discretionary Access Control, Mandatory Integrity Control, Authentication, Trusted Path, MLS

This Security Target (ST) has been prepared by BAE Systems Information Technology, LLC (hereinafter, BAE – IT), 2525 Network Place, Herndon, VA 22171. This ST supports the Target of Evaluation (TOE) as implemented within the XTS-400, Version 6.4(UKE) (hereafter referred to simply as Version 6.4(UKE)). Development of this ST was guided by specifications detailed in the following documents:

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, August 2005, Version 2.3
- Common Criteria for Information Technology Security Evaluation, Part 2: Security functional requirements, August 2005, Version 2.3
- Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance requirements, August 2005, Version 2.3.

1.2 Security Target Overview

This Security Target (ST) is for the BAE –IT supplied “XTS-400”, and specifies security requirements for obtaining an EAL5+ security evaluation. XTS-400 is a combination of an operating system, Secure Trusted Operating Program (STOP) revision 6, with BAE – IT-supplied hardware.

Operating systems evaluated against this ST will:

- Associate sensitivity labels with all objects and all its users will have an associated clearance level identifying the maximum security level of data that they may access.
- Allow simultaneous use of the system by multiple users, all with different clearances and needs-to-know.
- Allow simultaneous network connectivity to networks of differing sensitivities/classifications, including IPv6 networks.
- Provide mandatory integrity protection of files.
- Provide an untrusted operating environment that includes common Linux commands and tools.

- Provide an Application Programming Interface/Application Binary Interface which is suitable for running most Linux applications in their binary format (no recompilation required).

1.2.1 XTS-400 Background and History

XTS-200, XTS-300, STOP 3.1.E, STOP 3.2.E, STOP 4.1, and STOP 5.2.E are descendants of the Secure Communications Processor (SCOMP) [13], a system also developed by a predecessor of BAE –IT, which was evaluated by the National Computer Security Centre (NCSC) in 1984 and received an A1 rating. The hardware base is fundamentally different from the SCOMP's, and the software has undergone significant further development. Development of STOP 3.1 on the DPS6 PLUS began in 1987 and continued through 1989. During that time, the combination of the software and hardware became known as XTS-200.

In contrast to the hardware on which the SCOMP was based (the DPS6), the DPS6 PLUS and DPS 6000 incorporated virtual memory and ring-protection techniques from the Multics [12] system, so no additional hardware modification was required. The salient differences between the operating system employed by SCOMP and XTS-200 were that STOP 3.1 incorporated complete file system support within the TSF, that it was considerably restructured to improve its use of layering and other software engineering principles, that it was general-purpose, and that it provided a Unix-like untrusted operating environment. In addition, XTS-200 supported multiprocessor configurations. Development on STOP 3.2.E began in June 1992 and was completed in June 1993.

The essential differences between the XTS-300 and XTS-200 resulted from the change in hardware base, primarily to employ the functions provided by the Intel processor. Memory management, ring protection logic, and process management were updated in the XTS-300. In addition, I/O functions moved from TSF System Services (TSS) into the kernel. The user interface of the XTS-300 changed little from that of XTS-200. Development on STOP 4.1 began in November 1992 and was completed in October 1994. Development continued through STOP 5.2.E, which was completed in April 2000.

Development for the XTS-400 of STOP revision 6 began June 2000. STOP, revision 6 implements the same Application Program Interface as Red Hat® Linux® Version 8 (GLIB-C Version 6.0/2.1). Other changes to the OS were:

- Switch to the use of the more modern GCC Compiler
- Programs support is now in Ring 3 versus previous support in Ring 2
- Commodity Application System Services (CASS) is renamed Operating System Services (OSS) and now included in the TSF.

The XTS-400 with STOP 6 provides a general purpose computer system with a UNIX/Linux interface. The system includes trusted utilities to perform system maintenance and security administration functions. The XTS-400 is commodity software provided under scrupulous Configuration Management control. The system is distributed with an Installation and Setup Manual to show the user how to set up the hardware and with a Software Release Bulletin to show how to install/reinstall and configure the operating system. Additional user's manuals, both for operations and administrative functions, are also provided.

BAE –IT has other software products to augment the XTS-400 product. These other products are not part of the TOE addressed by this ST. Available as optional and separately priced items are:

- A Software Development Environment (SDE) package that allows programming of trusted and untrusted applications for use on the XTS. Frequently, initial programming and debug is done on a "real" Linux system and the binary copied to the XTS for execution. The SDE includes library functions to allow the security enforcing XTS API (separate from the Linux API used for UNIX functions).
- A middle-ware package called Secure Automated Guard Environment (SAGE) which provides transaction processing support for many of the tasks common to file-oriented filtering applications. SAGE reduces the risk and expense of developing custom applications by providing pre-written and pre-tested functions so the application developer can focus on the "security filter" logic.
- Turn-key applications programmed by BAE –IT to provide specific filtering capability.

1.3 CC Conformance Claim

The TOE is Part 2 conformant and Part 3 conformant EAL5 augmented with ALC_FLR.3 and ATE_IND.3 (Evaluation Assurance Level 5+), as defined in the "Common Criteria for Information Technology Security Evaluation Version 2.3" (CC). This ST conforms with applicable interpretations in effect as of 1 May 2007.

This ST claims conformance with the Certified Protection Profiles entitled "Labeled Security Protection Profile (Version 1.b)" and "Controlled Access Protection Profile (Version 1.d)", and is generally significantly more demanding in its functionality.

All international and UK interpretations that are final as of May 2007 shall apply to this ST. Interpretations that were superseded, too new, or not relevant to the CC itself, or not relevant to the requirements claimed in this ST shall not apply. Interpretations that affect the wording of this ST are marked with "*".

1.4 Conventions

The notation, formatting, and conventions used in this security target (ST) are consistent with version 2.3 of the Common Criteria for Information Technology Security Evaluation. Font style and clarifying information conventions were developed to aid the reader.

The CC permits four functional component operations: assignment, iteration, refinement, and selection to be performed on functional requirements. These operations are defined in Common Criteria v2.3, Part 1, as:

- Assignment: the specification of an identified parameter in a component;
- Refinement: the addition of details to a component
- Selection: the specification of one or more items from a list in a component
- Iteration: the use of a component more than once with varying operations.

This ST indicates which text is affected by each of these operations in the following manner:

- Assignments specified by the ST author are in **bold** text.
- Selections completed by the ST author are underlined.
- Refinements are identified with "Refinement:" right after the short name. Additions to the CC text are specified in **bold**. Deletions of the CC text are identified in the “End Notes” with a bold number after the component **(8)**.
- Iterations are identified with a number inside parentheses ("(#)"). These follow the short family name and allow components to be used more than once with varying operations.

In the case of an assignment contained within a selection the text will be both **bold** and underlined. For example, [selection: change, modify, [assignment: other options may apply]] would be represented as; modify, **create**.

Application Notes are used to provide the reader with additional requirement understanding or to clarify the author's intent. These are *italicized* and usually appear following the element needing clarification.

1.5 Terminology

This security target uses the terms described in this section to aid in the application of the requirements.

Access -A specific type of interaction between a subject and an object that results in the flow of information from one to the other.

Authorised administrator -An authorised user who has been granted the authority to manage the TOE. These users are expected to use this authority only in the manner prescribed by the guidance given to them.

Authorised user -A user who has been uniquely identified and authenticated. These users are considered legitimate users of the TOE.

Bell-Lapadula Security Model – security model for MAC enforced by the XTS-400 and detailed in; D.E. Bell and L.J. LaPadula, Secure Computer System: Unified Exposition and Multics Interpretation, ESD-TR-75-306, Electronic Systems Division (AFSC), 1976.

Biba – security model for MIC enforced by XTS-400 and detailed in; K.J. Biba, Integrity Considerations for Secure Computer Systems, MTR-3153, Mitre Corp., Bedford, Mass., April 1977.

Clearance -The users maximum authorization composed of a combination of sensitivity level and integrity level.

Component -The smallest selectable set of elements that may be included in a PP, an ST, or a package.

Critical Security Parameters (CSP) -Security-related information (e.g., authentication data such as passwords and pins) appearing in plaintext or otherwise unprotected form and whose disclosure or modification can compromise the security of the information protected by the module.

Discretionary Access Control (DAC) -A means of restricting access to objects based on the identity of subjects and/or groups to which they belong. The controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject.

Dominance – Concept in the Bell Lapadula security model. Dominance describes the comparison of a subject sensitivity level and object sensitivity level to determine whether sufficient privilege exists for the subject to access (i.e. dominate) the object.

Element -Individual requirements within a CC component; cannot be selected individually for inclusion in a PP, ST, or package.

Evaluation Assurance Level (EAL) -A package consisting of assurance components from CC, part 3 that represents a point on the CC predefined assurance scale.

Enclave -An environment that is under the control of a single authority and has a homogeneous security policy, including personnel and physical security and therefore protected from other environments. Local and remote elements that access resources within an enclave must satisfy the policy of the enclave. Enclaves can be specific to an organization or a mission and may contain multiple networks. They may be logical, such as an operational area network (OAN) or be based on physical location and proximity.

Greatest Lower Bound (GLB) – Concept in the Bell Lapadula security model. The GLB describes elements in a security label being compared with the lowest sensitivity level resulting.

Integrity label -A security attribute that represents the integrity level of a subject or an object. Integrity labels are used by the TOE as the basis for mandatory integrity control decisions.

Integrity level -The combination of a hierarchical level and an optional set of non-hierarchical categories that represent the integrity of the information.

Least Upper Bound (LUB) – Concept in the Bell Lapadula security model. The LUB describes elements in a security label being compared with the highest sensitivity level resulting.

Mandatory Access Control (MAC) -A means of restricting access to objects based on subject and object sensitivity labels. The Bell LaPadula model is an example of Mandatory Access Control.

Mandatory Integrity Control (MIC) -A means of restricting access to objects based on subject and object integrity labels. The Biba integrity model is an example of Mandatory Integrity Control.

Multilevel system (MLS) -A system that can simultaneously handle (e.g., share, process) multiple levels of data. It allows users at different sensitivity levels to access the system concurrently. The system permits each user to access only the data to which they are authorised access.

Named Object -An object that exhibits all of the following characteristics:

- The object may be used to transfer information between subjects of differing user identities within the TSF.
- Subjects in the TOE must be able to request a specific instance of the object.

- The name used to refer to a specific instance of the object must exist in a context that potentially allows subjects with different user identities to request the same instance of the object.

- The intended use of the object is for sharing of information across user identities.

Object -An entity within the TOE security functions scope of control (TSC) that contains or receives information and upon which subjects perform operations.

Operating Environment -The total environment in which an information system operates. It includes the physical facility and controls, procedural and administrative controls, and personnel controls.

Public Object -An object for which the TSF unconditionally permits all subjects “read” access. Only the TSF or privileged subjects may create, delete, or modify the public objects. No discretionary access checks or auditing is required for “read” accesses to such public objects. Attempts to create, delete, or modify such objects shall be considered security-relevant events, and, therefore, controlled and auditable. Objects that all subjects can read (e.g., the system clock) must be, implicitly, system low.

Protection Profile (PP) -An implementation-independent set of security requirements for a category of TOEs that meet specific consumer needs.

Security attributes -TSF data that contains information about subjects and objects and upon which access control decisions are based.

Security level -The combination of a hierarchical classification and a set of non-hierarchical categories that represent the sensitivity of the information.

Security Target (ST) -A set of security requirements and specifications to be used as the basis for evaluation of an identified TOE.

Sensitive information -Information that, as determined by a competent authority, must be protected because its unauthorised disclosure, alteration, loss, or destruction will at least cause perceivable damage to someone or something.

Sensitivity label -A security attribute that represents the security level of an object and that describes the sensitivity (e.g. Classification) of the data in the object. Sensitivity labels are used by the TOE as the basis for mandatory access control decisions.

Subject -An entity within the TSC that causes operations to be performed. Subjects can come in two forms: trusted and untrusted. Trusted subjects may be exempt from part or all of the TOE security policies. Untrusted subjects are bound by all TOE security policies.

Target of Evaluation (TOE) -An IT product or system and its associated administrator and user guidance documentation that is the subject of an evaluation.

TOE Security Functions (TSF) -A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TSP.

Unauthorised user -A user who may obtain access only to system provided public objects if any exist.

User -A term used to include both authorised and unauthorised entities (human user or external IT entity) outside the TOE that interacts with the TOE.

This page is intentionally left blank

Section 2 TOE Description

2.1 Product Description

This Security Target specifies Common Criteria requirements for the XTS-400, Version 6.4(UKE).

2.1.1 Software Overview

This section provides a general description of STOP, Version 6.4(UKE) STOP is a multitasking operating system.

The system supports both a mandatory sensitivity policy and a mandatory integrity policy. It provides 16 hierarchical sensitivity levels, 64 non-hierarchical sensitivity categories, eight hierarchical integrity levels, and 16 non-hierarchical integrity categories. Some of the hierarchical integrity levels are used by the system to provide role separation, and the others are available to users. The combination of mandatory sensitivity hierarchical and non-hierarchical levels is called the Mandatory Access Control (MAC) label. The combination of mandatory integrity hierarchical and non-hierarchical levels is called the Mandatory Integrity Control (MIC) label. The system also supports a discretionary access control policy.

2.1.1.1 Software Components

The Security Kernel software occupies the innermost and most privileged ring and performs all Mandatory Access Control (MAC), and Mandatory Integrity Control (MIC). The kernel provides a virtual process environment, which isolates one process from another. The kernel implements a variation of the reference monitor concept. When a process requests access to an object, the kernel performs the access checks, and, given that the checks pass, maps the object into the process' address space. Subsequent accesses are mediated by the hardware. The Security Kernel also provides I/O services and an Inter-process Communication (IPC) message mechanism. The Security Kernel is part of every process' address space and is protected by the ring structure supported by the hardware.

The TSS software executes in Ring 1. TSS provides trusted system services required by both trusted and untrusted processes. The Kernel, TSS and OSS have the responsibility for creation and loading of both trusted and untrusted programs, respectively, in XTS-400, Version 6.4(UKE). TSS software enforces the Discretionary Access Control (DAC) policy to file system objects.

OSS executes in Ring 2. OSS provides a Linux interface for user-written and trusted and untrusted software applications. The purpose of OSS is to make the multilevel security execution environment hidden to software running in the Application Domain (Ring 3).

Ring 3 is the Application Domain, in which all applications, both trusted and untrusted, execute. Software is considered trusted in XTS-400, Version 6.4(UKE) if it performs functions upon which the system depends to enforce the security policy (e.g., the establishment of user authorization). This determination is based on integrity level and privileges. Untrusted software runs at integrity

level 3, with all integrity categories, or lower. Some processes require privileges to perform their functions. An example of a process that requires privileges is the Secure Server, which needs access to the User Access Authentication database, kept at system high access level, while establishing a session for a user at another security level.

The multilevel secure XTS-400, Version 6.4(UKE) is designed to provide a high level of security for many environments, which includes applications that may filter the information according to rules based upon the security policies required by the site. The system supports both a mandatory sensitivity policy and a mandatory integrity policy. It provides hierarchical sensitivity levels, non-hierarchical sensitivity categories, hierarchical integrity levels, and non-hierarchical integrity categories. The system provides for user identification and authentication used for policy enforcement through user identifiers and passwords, and individual accountability through its auditing capability. Data scavenging is prevented through the control of object reuse. The trusted path mechanism is provided by the implementation of a Secure Attention Key (SAK). The separation of administrator and operator roles is enforced through integrity protected operations.

2.1.2 Hardware Overview

The TOE includes only standard-PC, commercial off-the-shelf (COTS) components. The hardware configuration put forward for evaluation is the “Model 3200”. The Model 3200 is built around an Intel XEON processor and Intel SE7520BD2SCSI motherboard. There are different form factor solutions (tower, 6U, 5U, 3U, 2U, etc.) and optional add-on hardware. Optional add-on hardware is also detailed in this section.

2.1.3 Minimal Evaluated Configuration

Component	**Specific Kinds Included in the Evaluation**	**Notes**
CPU	INTEL Xeon (P4) “Prestonia” “Nocona”	Intel IA-32 architecture, *Intel Architecture Software Developer’s Manual* (Volumes 1 to 3)
Hard Disk (SCSI-3, ANSI X3T9.2/85-52)	Seagate “Cheetach”	Additional hard disks can optionally be connected: only 68-pin, SCSI LVD connections are used
Motherboard	Intel SE7520BD2SCSI	Includes: ▪ PCI 2.1 or 2.2 system bus ▪ 512Mbytes – 2Gbytes of main memory ▪ up to two 16550-compatible UARTs (serial controllers) ▪ 8254-compatible PIT

Component	Specific Kinds Included in the Evaluation	Notes
		▪ MC146818 – compatible RTC ▪ 8259-compatible PIC ▪ 8237-compatible DMA ▪ 87077A-compatible floppy controller ▪ IDE ATA-100 compatible disk controller
Floppy Drive	3.5” IDE 1.44MB Microfloppy disk drive	
SCSI Adapter	Adaptec 29160	
Tape Drive	Wangdat Tecmar TS3400-D01, HP C1554C, HP C5686C	SCSI 4mm, ANSI X3T9.2/85-52, DAT DDS-2, DAT DDS-3, DAT DDS-4
Video Controller	Cirrus Logic CL-GD5480, ATI Rage XL	
CD-ROM/DVD Drive (ANSI X3T9.2/85-52)	Toshiba XXM6401B, Toshiba SD-M1711S, Lite-on LH-52C1P	Lite-on has ATAPI interface while the others are SCSI
Monitor	VGA, SVGA. CRT or flat panel.	
Keyboard	101-key or 104-key standard or laptop style, PS/2	<SysRq> key must generate scancode 84
Mouse or Touchpad	PS/2	
PCI Parallel Port	IEEE 1284 ECP parallel controller	

The optional add-on hardware is:

Component	**Specific Kinds Included in the Evaluation**	**Notes**
PC Card Reader (PCMCIA)	Adtron SDDS Litronic 2108	Multiple can be connected
Gigabit Ethernet Controller	Intel PRO/1000	On motherboard or part of plug-in PCI card; up to 4 cards per system
2/4 Port 10/100 Ethernet Card (IEEE 802.3)	Zynx ZX-34nQ, ZX-37n	Up to 4 cards per system
Printer	HP 4000 series (supports PCL-5) Any ASCII-only printer that: ▪ does not support graphics modes ▪ does not support special initialisation sequences ▪ uses XON/XOFF flow control	Parallel connection only; includes 4100 and 4200 sub-series
Serial Terminal		Must meet the requirements detailed in the section *Serial Terminal Characteristics* below.

The processor incorporates its own ring protection mechanism supporting four rings, descriptor privilege levels, gate descriptors, segment attributes (read, write, execute), and call/return instruction. The privilege level (PL) protection mechanism ranges from PL0 (the most privileged) to PL3 (the least privileged).

STOP supports only single-processor hardware with Hyper-Threading disabled.

Other hardware that is not part of the TOE, but is an optional part of an XTS-400 system includes

- APC Smart-UPS uninterruptible power supply
- Mission Support Cryptographic Unit (MSCU).

This optional hardware has been reviewed by BAE-IT, but is not part of this evaluation. The MSCU is a proprietary PCI board that supports “type 1” cryptography and has been separately scrutinised by the U.S. National Security Agency. The software to support the MSCU is part of the TOE and the evaluation.

Power, cooling, mechanical, and form factors are irrelevant.

2.1.3.1 Serial Terminal Characteristics

Any terminal connected to the XTS-400 system must provide, at a minimum, a keyboard for providing input to the system, and a video display or printing device for transmitting system output to the user in human-readable form. A video display must be capable of displaying at least 24 lines at one time. The terminal must operate in full duplex i.e. non-local echoing mode.

Any terminal connected to the XTS-400 system must provide a correct association between the user’s actions of depressing keyboard keys and transmitting the ASCII representations of those keys via the communication interface. Any terminal connected to the XTS-400 system must provide a correct association between the ASCII characters transmitted from the system and their displayed or printed representations.

Any terminal connected to the XTS-400 system must provide a direct means of generating a serial BREAK signal through the terminal keyboard. Also, it must be impossible for the terminal to generate the BREAK signal in response to any sequence of signals sent from the system to it, and impossible for any sequence of signals sent from the system to inhibit the terminal’s ability to generate the BREAK signal from the keyboard.

Any terminal connected to the XTS-400 system must provide a means of setting either the Data Set Ready or the Carrier Detect signals, and a means of clearing both of these signals. This can be provided either by the terminal itself (such as turning the terminal’s power off and on) or by an external mechanism between the terminal and the XTS-400 system.

Any terminal connected to the XTS-400 system must provide a means for the terminal user to clear any internal store e.g. screen display, programmable function keys, buffer memory or external store e.g. printer paper, local disk between user sessions.

2.1.4 TOE Definition

The TOE consists of:

- the Kernel, TSS, OSS, and Trusted Software components mentioned above;
- some BAE -IT-written untrusted software to ease use of the untrusted environment;
- some third-party untrusted software that is shipped with XTS systems to customers by BAE -IT to ease installation by the customer and to provide the look and feel of a Linux system;
- BIOS software to perform certain kinds of hardware configuration or diagnostics;
- hardware, as defined in the previous section.

The TOE also includes the optional XTS-400 AutoStart feature. This feature activates code that is already part of the TSF.

The TOE is not a distributed system, though it can be attached to multiple Ethernet networks concurrently.

The TOE is not an embedded system.

The TOE does not provide a particular trusted application out-of-the-box, but is a general-purpose system that can support many kinds of highly trusted applications. BAE –IT and its customers have written a number of trusted applications which rely on the security features provided by the XTS-400.

The TOE is based on standard PC hardware, but can support concurrent use by multiple users and can support multiple concurrent printer, terminal, and network connections.

2.2 General TOE Functionality

The conformant operating system includes the following security features:

2.2.1 Security Features

- Identification and Authentication which mandates authorised users to be uniquely identified and authenticated before accessing information stored on the system;
- Discretionary Access Control (DAC) which restricts access to objects based on the identity of subjects and/or groups to which they belong, and allows authorised users to specify protection for objects that they control;
- Mandatory Access Control (MAC) which enforces the data sensitivity classification model (i.e., Unclassified, Restricted, Confidential, Secret, Top Secret) on all authorised users and all TOE resources;
- Mandatory Integrity Control (MIC) which enforces an integrity policy on all authorised users and TOE resources to prevent malicious entities from corrupting data;
- Audit services which allow authorised administrators to detect and analyze potential security violations.
- Trusted path, which allows a user to be sure s/he is interacting directly with the TSF during sensitive operations.
- Isolation, of the TSF code and data files from the activity of untrusted users and processes.
- Separation, of processes from one another (so that one process/user can not tamper with the internal data and code of another process).

2.2.2 Other Characteristics of the TOE

- the ability to process multiple security levels of information in a multilevel environment,

- an untrusted operating environment that includes most common Linux commands and tools,
- a programming API that allows most common Linux applications to run “out-of¬the-box”,
- simultaneous network connectivity to at least 8 LANs,
- the inability to provide mechanisms or services to ensure availability of data residing on the TOE. [If availability requirements exist, the environment must provide the required mechanisms (e.g., mirrored/duplicated data)].

This page is intentionally left blank

Section 3 TOE Security Environment

3.1 Secure Usage Assumptions

A.Acc_to_Comms: Physical protection of communications. Physical protection of the communications to the system is adequate to guard against unauthorised access or malicious modification by users.

A.Admin_Docs: Documentation for administrators. System Administrators follow the policies and procedures defined in the TOE documentation for secure administration of the TOE.

A.Admin_Errors: Potential for administrator errors. System administrators are fallible and occasionally make errors that compromise security.

A.Clearance: Authorisation procedures. Procedures exist for granting users authorization for access to specific security levels. This includes procedures for establishing one or more operators and administrators.

A.Competent_Admin: Competent system administrators. System administrators are competent to manage the TOE and the security of the information it contains.

A.Coop_User: Cooperative users. Users cooperate with those responsible for managing the TOE to maintain TOE security, follow TOE user guidance, protect TOE secrets, and follow site procedures.

A.Dispose_User_Data: Disposal of user data. System Administrators properly dispose of user data after access has been removed (e.g., due to job termination, change in responsibility).

A.Handling_of_Data: Data handling procedures. Procedures exist for how sensitive, classified, and high-integrity data and secrets are to be handled when they are in the possession of an authorised user. Procedures also exist for pick-up and distribution of hardcopy output at multi-user or multi¬level printers.

A.Outsider_Hi: Expert threat agents. The TOE is subject to deliberate attack by experts with advanced knowledge of security principles and concepts employed by the TOE. These experts are assumed to have substantial resources and high motivation.

A.Password_Management: Password management promoting user compliance. System Administrators follow password management policies and procedures to ensure users comply with password policies.

A.Phys_Acs_to_Out: Physical access. The TOE is located within controlled access facilities that prevent unauthorised physical access by outsiders.

A.Protect_From_Out: TOE protection from outsiders. The TOE involved in security policy enforcement will be physically protected from unauthorised modification by potentially hostile outsiders.

A.Review_Audit_Log: Administrators review audit logs. System Administrators review audit logs regularly.

A.Secure_Term: Terminal procedures. Procedures exist for how to restrict other system users, or non-system users, from viewing terminal output on an authorised user's terminal. This includes considerations such as "looking over the shoulder", an authorised user leaving his or her terminal unattended, and terminal-specific instructions to erase terminal-local data following a logout.

A.Sensitivity: Procedures for setting levels and marking. Procedures exist for establishing the security attributes of all information imported into the system, for establishing the security attributes for all peripheral devices (e.g., printers, tape drives, disk drives) attached to the TOE, and marking a sensitivity label on all hardcopy output generated.

A.User_Mistakes: Mistakes by users. Users are fallible and occasionally make errors that compromise security.

A.Physical Secure: Physical Location. It is assumed that appropriate physical security is provided within the environment for the protective marking of the IT assets protected by the operating system and the protective marking of the stored, processed, and transmitted information.

A.No_Trusted_Network: The physical connection of another system to the XTS-400 is assigned appropriate MAC and MIC levels and is therefore trusted and under the management regime of the XTS400.

3.2 Security Threats

The asset being protected from the threats detailed in the table below is the user data processed and hosted on the XTS-400 server. Specifically, these threats may compromise the confidentiality, integrity or availability of this asset.

Threat agents may be authorised users of the TOE with legitimate access to the asset being protected or unauthorised users with no legitimate access to the asset being protected. Both types of threat agents may bring about hostile or accidental threats as indicated in the threat description.

Table 1: Security Threats

Threat Reference	**Threat Agent**	**Threat Agent: Resources Available and Motivation**	**Threat Description**
T.Admin_Err_Commit: Administrative errors of commission.	Authorised Administrator	High resources Low motivation	An administrator commits errors that directly compromise organisational security objectives or change the technical security policy enforced by the system or application.

Threat Reference	Threat Agent	Threat Agent: Resources Available and Motivation	Threat Description
T.Admin_Err_Omit: Administrative errors of omission.	Authorised Administrator	High resources Low motivation	The system administrator fails to perform some function essential to security. This could leave user data open to unauthorised disclosure, modification, or deletion; could allow an attack in progress to continue (which could jeopardize the TSF itself); or could allow a user to continue to use the system with a clearance or capabilities that have been determined to be too lenient.
T.Admin_Hostile_Modify: Hostile administrator modification of user or system data.	Authorised Administrator	High resources High motivation	An administrator maliciously obstructs organizational security objectives or modifies the system's configuration to allow security violations to occur.
T.Covert_Channel:	Hacker	Low, medium or high resources High motivation	Covert channel allows MAC bypass. A hacker bypasses the MAC policy by use of covert channels. This could lead to disclosure of sensitive user data to unauthorised users.
T.Dev_Flawed_Code: Software containing security-related flaws.	Authorised developer	High resources Low motivation	A system or applications developer delivers code that does not perform according to specifications or contains security flaws. This could lead to arbitrary compromise of user and TSF data, corruption of the TSF itself, and unauthorised use of the system.
T.Hack_AC: Hacker undetected system access.	Hacker	Low, medium or high resources High motivation	A hacker gains undetected access to a system due to missing, weak and/or incorrectly implemented access control causing potential violations of integrity, confidentiality.
T.Hack_Phys: Exploitation of vulnerabilities in the physical environment of the system.	Hacker	Low, medium or high resources High motivation	A hacker physically interacts with the system to exploit vulnerabilities in the physical environment, resulting in arbitrary security compromises.
T.Hack_Social_Engineer: Social engineering.	Hacker	Low, medium or high resources High motivation	A hacker uses social engineering techniques to gain information about system entry, system use, system design, or system operation.

Threat Reference	Threat Agent	Threat Agent: Resources Available and Motivation	Threat Description
T.Insecure_Start: Insecure startup after a system crash.	Potential system vulnerability	N/A	After a system crash, the information used by the system is in an inconsistent state that causes a TOE malfunction or allows circumvention of policy.
T.Malicious_Code: Malicious code exploitation.	Authorised User / hacker	Various	An authorised user, IT system, or hacker downloads and executes malicious code, which causes abnormal processes that violate the integrity or confidentiality of system assets.
T.Objects_Not_Clean: Residual data left in object.	Potential system vulnerability	N/A	Residual information is left in an object and when reallocated to another subject, the information is still viewable.
T.Power_Disrupt: Unexpected disruption of system or component power.	Potential system vulnerability	N/A	A human or environmental agent disrupts power causing the system to lose information or security protection.
T.Spoofing: Legitimate system services are spoofed.	Authorised Administrator or Hacker	Various	An attacker tricks users into interacting with spurious system services. This could allow disclosure of the users' data to the attacker, unauthorised modification of the users' data by the attacker, and general circumvention of the user identification and DAC mechanisms.
T.User_Improper_Export: Hostile user acts cause confidentiality breaches.	Authorised user	Medium resources (can only remove data to which they have access rights) High motivation	A user collects sensitive or proprietary information and removes it from the system.
T.User_Abuse: User abuses authorisations.	Authorised user	Medium resources (can only remove data to which they have access rights) Low motivation	User abuses granted authorizations to improperly collect sensitive or security-critical data; to improperly change or destroy sensitive or security-critical data; or to improperly send sensitive or security-critical data.

Threat Reference	Threat Agent	Threat Agent: Resources Available and Motivation	Threat Description
T.User_Error: User errors cause security breaches.	Authorised user	Low resources Low motivation	A user commits errors that cause information to be delivered to the wrong place or wrong person; accidentally deletes user data or changes system data rendering user data inaccessible; or commits errors that induce erroneous actions by the system and/or erroneous statements by its users.
T.Trusted_User_Error: Trusted user errors undermine the system's security features.	Authorised user	Low resources Low motivation	A trusted user commits errors that cause the system or one of its applications to undermine the system's security features.
T.Audit_Corrupt: Corrupt Audit Records.	Hacker	Various resources High motivation	A malicious process may cause audit records to be lost or modified, or prevent future records from being recorded by taking actions to exhaust audit storage capacity, thus masking an attacker’s actions.
T.Config_Corrupt: Corrupt Configuration Data.	Hacker	Various resources High motivation	A malicious process may cause configuration data or other trusted data to be lost or modified.
T.Improper_Installation: Installation Errors.	Authorised developer/user	High resources Low motivation	Operating system may be delivered insecurely by the developer, or installed or configured, by the operator, in a manner that undermines security.
T.Poor_Design: Design Errors.	Authorised developer	Low resources Low motivation	Unintentional or intentional errors in requirement specification, design or development of the IT operating system, on the part of the developer, may occur.
T.Poor_Implementation: Implementation Errors.	Authorised developer	High resources Various motivation	Unintentional or intentional errors in implementing the design of the IT operating system, on the part of the developer, may occur.

Threat Reference	Threat Agent	Threat Agent: Resources Available and Motivation	Threat Description
T.Poor_Test: Inadequately Tested.	Potential system vulnerability	N/A	Incorrect system behaviour may result from inability, on the part of the system operator, to demonstrate that all functions and interactions within the operating system operation are correct.
T.Replay: Replay.	Authorised user or hacker	Various	A malicious process or user may gain access by replaying authentication (or other) information. This could allow unauthorised use of the system and allow unauthorised access to user data (as the attacker could operate with the identity and clearance of the user whose authentication information was copied).
T.Sysacc: Unauthorised Admin Access.	Authorised user or hacker	Various resources High motivation	A malicious process or user may gain unauthorised access to the administrator account, or that of other trusted personnel. This could allow unauthorised use of the system and unauthorised access to both user and TSF data (as the attacker could operate with the identity and clearance of the trusted user whose account was used).
T.Unattended_Session: Unauthorised Session Access.	Authorised user or hacker	Various High motivation	A malicious process or user may gain unauthorised access to an unattended session. This could allow unauthorised use of the system and allow unauthorised access to user data (as the attacker could operate with the identity and clearance of the user whose session was taken).
T.Unauth_Modification: Unauthorised Modifications.	Authorised user or hacker	Various High motivation	Unauthorised modification or use of IT operating system attributes and resources, by a malicious user or process, may occur.
T.User_Corrupt: Corrupt User Data.	Authorised users	Low resources Low motivation	User data may be lost or tampered with by other users.

3.3 Organisational Security Policies

P.Accountability: Individual Accountability. Individuals shall be held accountable for their actions.

P.Authorities: Authorities Notified of Threats. Appropriate authorities shall be immediately notified of any threats or vulnerabilities impacting systems that process their data.

P.Authorised_Use: Authorised Use of Information. Information shall be used only for its authorised purpose(s).

P.Authorised_Users: Authorised Use of System. Only those users who have been authorised to access the information within the system may access the system.

P.Availability: Availability of Information. Information shall be available to satisfy mission requirements.

P.Classification: Access must be limited by sensitivity. The system must limit the access to information based on sensitivity, as represented by a label, of the information contained in objects, and the formal clearance of users, as represented by subjects, to access that information. The access rules enforced prevent a subject from accessing information which is of higher sensitivity than it is operating at and prevent a subject from causing information from being downgraded to a lower sensitivity.

P.Guidance: Installation and Use Guidance. Guidance shall be provided for the secure installation and use of the system.

P.Information_AC: Information Access. Information shall be accessed only by authorised individuals and processes.

P.Integrity: Information Integrity. Information shall retain its content integrity.

P.Marking: Information Marking and Labeling. Information, and other system assets such as hard drives, printers etc., shall be appropriately marked and labeled.

P.Need_to_Know: Users must have a need to know. The system must limit the access to and modification of the information in protected resources to those authorised users which have a “need to know” for that information.

P.Physical_Control: Information Physically Protected. Information shall be physically protected to prevent unauthorised disclosure, destruction, or modification.

This page is intentionally left blank

Section 4 Security Objectives

Listed below are the objectives for the TOE and for the IT environment. Objectives marked with "*" map to objectives of the LSPP, to which this ST claims conformance.

4.1 Security Objectives for the TOE

O.AC_Admin_Limit: Limitation of administrative access control. Design administrative functions in such a way that administrators do not automatically have access to user objects, except for necessary exceptions. For an accounts administrator, the necessary exceptions include allocation and de-allocation of storage. For an audit administrator, the necessary exceptions include observation of audited actions. In general, the exceptions tend to be role specific.

O.Audit_Gen_User*: Individual accountability. Provide individual accountability for audited events. Uniquely identify each user so that auditable actions can be traced to a user.

O.Access*: User authorised access. The IT operating system will ensure that users gain only authorised access to it and to its resources that it controls.

O.Access_History: Authorised user access history. The system will display information (to authorised users) related to previous attempts to establish a session.

O.Admin_Role: Isolate administrative actions. The operating system will provide an administrator role to isolate administrative actions.

O.Admin_Trained*: Trained administrators. The IT operating system will provide authorised administrators with the necessary information for secure management.

O.Audit_Generation*: Audit generation. The IT operating system will provide the capability to detect and create records of security relevant events associated with users.

O.Audit_Protection: Protect audit. The IT operating system will provide the capability to protect audit information.

O.Audit_Review*: Selectively view audit. The IT operating system will provide the capability to selectively view audit information.

O.Covert_Channel_Reduction: The TSF will minimise the covert channels that exist to circumvent the MAC policy.

O.Discretionary_Access*: Discretionary access control. The IT operating system will control accesses to resources based upon the identity of users and groups of users.

O.Discretionary_User_Control*: Discretionary user access. The IT operating system will allow authorised users to specify which resources may be accessed by which users and groups of users.

O.Display_Banners: Sensitivity banner display. The system will display an advisory warning regarding use of the TOE.

O.Manage*: Administrator support management. The IT operating system will provide all the functions and facilities necessary to support the authorised administrators in their management of the security of the IT system.

O.Mandatory_Access*: Mandatory access control. The IT operating system will control accesses to resources based upon the security levels of users and resources and the “need to know” of the users.

O.Mandatory_Integrity: Mandatory integrity control. The IT operating system will control accesses to resources based upon the integrity levels of users and resources and the “need to know” of the users.

O.Markings: Sensitivity markings. The IT operating system will provide the capability to mark printed output with accurate sensitivity labels.

O.Protect: Data and resource protection. The IT operating system will provide means to protect user data and resources.

O.Recovery: Assured recovery. Procedures and/or mechanisms will be provided to assure that recovery is obtained without a protection compromise, such as from system failure or discontinuity.

O.Residual_Information*: Residual information. The IT operating system will ensure that any information contained in a protected resource is not released when the resource is reallocated.

O.Self_Protection: Self protecting system. The operating system will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorised disclosure.

O.Trained_Users:User training. The IT operating system will provide authorised users with the necessary guidance for secure operation.

O.Trusted_Path: Trusted path. The operating system will provide a means to ensure users are not communicating with some other entity pretending to be the operating system.

O.Trusted_System_Operation*: Trusted system operation. The IT operating system will function in a manner that maintains IT security.

O.User_Authentication Verify user identity. The operating system will verify the claimed identity of the user.

O.User_Identification: Unique user identity. The operating system will uniquely identify users.

4.2 Security Objectives for the Environment

OE.Admin: Administrators competent. Individuals given the role of administrator are competent with regard to managing the TOE and the security aspects of the users and information associated with the TOE. These individuals understand the TOE administrative guidance and site-specific procedures and follow that guidance and those procedures.

O.Authorities: Authorities notified of threats. The system vendor will have a flaw remediation mechanism in place such that appropriate authorities shall be notified of any threats or vulnerabilities impacting the TOE.

OE.Clearance: Procedures to establish authorisations. Procedures will be provided to determine the "trust" (in terms of MAC and MIC clearance, group membership, XTS capabilities, and access to the console), and needs-to-know, for each system user.

O.Config_Mgmt*: Configuration management. All changes to the operating system and its development evidence will be tracked and controlled.

OE.Handling_of_Data*: Data handling procedures. Procedures will be provided for how sensitive, classified, and high-integrity data and secrets are to be handled when they are in possession of an authorised user. Procedures also will be provided for pick-up and distribution of hardcopy output at multi-user or multi-level printers.

O.Install*: Installation guidance. The IT operating system will be delivered with the appropriate installation guidance to establish and maintain IT security.

OE.Connection_Mgmt: Physical connection management. If the TOE is connected to another system, the components of the physical connection must under common management to ensure that the physical connection to the TOE is at a particular MAC and MIC level.

OE.Physical*: Physical security. Physical security will be provided within the environment according to the protective marking of the IT assets protected by the operating system and the protective marking of the stored, processed, and transmitted information.

OE.Secure_Term: Terminal procedures. Procedures will be provided for how to restrict other system users, or non-system users, from viewing terminal output on an authorised user's terminal. This includes considerations such as "looking over the shoulder", an authorised user leaving his or her terminal unattended, and terminal-specific instructions to erase terminal-local data following a logout.

OE.Sensitivity: Procedures for setting levels and marking. Procedures will be provided for establishing the security attributes of all information imported into the system, for establishing the security attributes for all peripheral devices (e.g., printers, tape drives, disk drives) attached to the TOE, and marking a sensitivity label on all hardcopy output generated.

O.Sound_Design*: Practice sound design. The design of the IT operating system will be the result of sound design principles and techniques, which are accurately documented.

O.Sound_Implementation*: Sound implementation. The implementation of the IT operating system will be an accurate instantiation of its design.

O.Testing: Testing of TOE code. The IT operating system will be tested functionally, with moderate levels of depth and coverage, tested independently, and subjected to penetration testing.

OE.User: Users competent. Individuals given access to the TOE are competent with regard to understanding the TOE user guidance and site-specific procedures and can follow the guidance and procedures. Users given additional clearance, group membership, capabilities, or access to the console are competent to understand the security ramifications and competent to use those "powers" safely.

O.Vulnerability_Analysis: Vulnerability analysis. The TOE shall undergo a comprehensive vulnerability analysis as described in the assurance requirements.

This page is intentionally left blank

Section 5 IT Security Requirements

This section defines the functional requirements within the ST that are the basis of the TOE Functional Requirements. The sub-sections contained in this section will define each functional requirement from the CC part 2 as they relate to the TOE. In some cases, the requirements allow for assignment, selection, or refinement of specific parameters or variables. The assignments, selections, and refinements (and iterations) of these requirements are represented as described in the *Conventions* section of this document.

These security functional requirements are summarised in Table 5-1. Requirements that are part of the LSPP (to which this ST claims conformance) are denoted with "*" in the table.

Table 2: Security Functional Requirements

Class Name	Functional Family	Dependencies	Hierarchical to
Security Audit	FAU_GEN.1*	FPT_STM.1	None
	FAU_GEN.2*	FAU_GEN.1, FIA_UID.1	None
	FAU_SAA.1	FAU_GEN.1	None
	FAU_SAR.1*	FAU_GEN.1	None
	FAU_SAR.2*	FAU_SAR.1	None
	FAU_SAR.3*	FAU_SAR.1	None
	FAU_SEL.1*	FAU_GEN.1, FMT_MTD.1	None
	FAU_STG.2	FAU_GEN.1	FAU_STG.1*
	FAU_STG.3*	FAU_STG.1	None
	FAU_STG.4*	FAU_STG.1	FAU_STG.3
User Data Protection	FDP_ACC.2	FDP_ACF.1	FDP_ACC.1*
	FDP_ACF.1*	FDP_ACC.1, FMT_MSA.3	None
	FDP_ETC.1*	{FDP_ACC.1 or FDP_IFC.1}	None
	FDP_ETC.2*	{FDP_ACC.1 or FDP_IFC.1}	None
	FDP_IFC.2	FDP_IFF.1	FDP_IFC.1*
	FDP_IFF.2*	FDP_IFC.1, FMT_MSA.3	FDP_IFF.1

Class Name	Functional Family	Dependencies	Hierarchical to
	FDP_ITC.1*	FMT_MSA.3, {FDP_ACC.1 or FDP_IFC.1}	None
	FDP_ITC.2*	{FDP_ACC.1 or FDP_IFC.1}, {FTP_ITC.1 OR FTP_TRP.1}	None
	FDP_RIP.2*	None	FDP_RIP.1
Identification and Authentication	FIA_AFL.1	FIA_UAU.1	None
	FIA_ATD.1*	None	None
	FIA_SOS.1*	None	None
	FIA_UAU.2	FIA_UID.1	FIA_UAU.1*
	FIA_UAU.4	None	None
	FIA_UAU.7*	FIA_UAU.1	None
	FIA_UID.2	None	FIA_UID.1*
	FIA_USB.1*	FIA_ATD.1	None
Security Management	FMT_MOF.1	FMT_SMR.1, FMT_SMF.1	None
	FMT_MSA.1*	FDP_ACC.1, FDP_IFC.1, FMT_SMR.1, FMT_SMF.1	None
	FMT_MSA.2	ADV_SPM.1, FMT_MSA.1, FMT_SMR.1 {FDP_ACC.1 or FDP_ICF.1}	None
	FMT_MSA.3*	FMT_MSA.1, FMT_SMR.1	None
	FMT_MTD.1*	FMT_SMR.1, FMT_SMF.1	None
	FMT_MTD.2	FMT_MTD.1, FMT_SMR.1	None
	FMT_REV.1*	FMT_SMR.1	None
	FMT_SAE.1	FMT_SMR.1, FPT_STM.1	None
	FMT_SMF.1	None	None

Class Name	Functional Family	Dependencies	Hierarchical to
	FMT_SMR.1*	FIA_UID.1	None
	FMT_SMR.3	FMT_SMR.1	None
Protection of the TOE Security Functions	FPT_AMT.1*	None	None
	FPT_RCV.1	FPT_TST.1, AGD_ADM.1, ADV_SPM.1	None
	FPT_RVM.1*	None	None
	FPT_SEP.1*	None	None
	FPT_STM.1*	None	None
	FPT_TST.1	FPT_AMT.1	None
TOE Access	FTA_SSL1	FIA_UAU.1	None
	FTA_SSL2	FIA_UAU.1	None
	FTA_SSL.3	None	None
	FTA_TAB.1	None	None
	FTA_TAH.1	None	None
	FTA_TSE.1	None	None
Trusted Path/Channels	FTP_TRP.1	None	None

5.1 TOE Security Functional Requirements

5.1.1 5.1.1 Security audit (FAU)

5.1.1.1 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 basic level of audit;

c) All auditable events listed in Appendix F.

Application Note: Records of the start-up and shutdown of audit functions can be indirect, e.g., the system startup and shutdown records imply start and stop of auditing. It is assumed that auditing continues to function even through system reboots and has to be actively disabled. Enabling and disabling of audit functions can be marked with a record showing use of an administrative command that performs those functions. Enabling and disabling of the audit functions may not actually take effect until the system is rebooted.

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 (if applicable), and the outcome (success or failure) of the event;

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, **the MAC (sensitivity) and MIC levels of the process, effective owner and group of the process, effective privileges of the process; and the following specific information for the following kinds of events**:

Audit Event	Additional Information in Audit Record
Those associated with particular objects or information	**the sensitivity level of the object or information**
All requests to perform an operation on an object covered by the SFP (FDP_ACF.1)	**the identity of the object**
All use of the user identification mechanism, including the identity provided during successful attempts (FIA_UID.2)	**The origin of the attempt (e.g., terminal identification)**
All modifications of the values of TSF data, including audit data (FMT_MTD.1)	**The old and new values of the TSF data**
Modifications to the group of users that are part of a role and every use of the rights of a role (FMT_SMR.3)	**The role and origin of the request**

5.1.1.2 User identity association (FAU_GEN.2)

FAU_GEN.2.1 **Refinement: For audit events resulting from actions of identified users**, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

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 **failed login attempts** known to indicate a potential security violation;

b) No additional rules

5.1.1.4 Audit review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide **authorised administrator** with the capability to read **Size of the audit record; Type of event being audited; Time the audit record is generated; Process ID of the process causing the audit event; MAC label of the process; Effective privileges of the process; Real user ID; Real group ID** from the audit records.

FAU_SAR.1.2 **Refinement:** The TSF shall provide the audit records in a manner suitable for the authorised administrator to interpret the information **using a tool to access the audit trail.**

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 Selectable Audit review (FAU_SAR.3)

FAU_SAR.3.1 The TSF shall provide the ability to perform searches of audit data based on **the following attributes**:

a) audit event

b) device name

c) file name

d) process ID

e) start date

f) stop date

g) trusted editor name

h) trusted editor request

i) user ID

j) group ID

k) semaphore ID

l) shared memory ID

m) object MAC (sensitivity) and MIC (integrity) level

n) subject MAC and MIC sensitivity level.

Application Note: Attribute selection may be any or all that are searchable.

5.1.1.7 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, user identity;

b) object MAC (sensitivity) and MIC (integrity) level;

c) subject MAC and MIC level.

5.1.1.8 Guarantees of audit data availability (FAU_STG.2)

FAU_STG.2.1 **Refinement**: The TSF shall protect the stored audit records **in the audit trail** 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.

Application Note: When audit storage exhaustion occurs, the system may shutdown, or with operator notice, transition to a maintenance mode of operation.

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

FAU_STG.3.1 The TSF shall take **generate an alarm to the authorised administrator** if the audit trail exceeds a **configured number of recent audit files**.

Application Note: The alarm may be a text message to the system console. An administrator can configure a value for the number of new audit files.

5.1.1.10 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 and **cause the system to shutdown** if the audit trail is full.

Application Note: Disk space available for audit files is based upon the size of the hard drive.

5.1.2 User data protection (FDP)

5.1.2.1 Complete Access Control (FDP_ACC.2)

FDP_ACC.2.1 The TSF shall enforce the **Discretionary Access Control policy** on **all subjects and all named objects** and all operations among subjects and objects covered by the SFP.

Application Note: The TOE supports only one subject type: a process (see Appendix B). A discussion of TOE supported objects is in Appendix C.

FDP_ACC.2.2 The TSF shall ensure that all operations between any subject in the TSC and any object within the TSC are covered by an access control SFP.

5.1.2.2 Access Control Functions (FDP_ACF.1)

FDP_ACF.1.1 The TSF shall enforce the **Discretionary Access Control policy** to objects based on the following:

a) For all subjects and objects:

- **owner's UID and the allowed and denied operations for that user;**
- **owner's group and the allowed and denied operations for that group;**
- **other listed users and/or groups and the allowed and denied operations for those users and/or groups; and**
- **the allowed and denied operations for other unlisted users and/or groups**

b) For all subjects, the "real" (originating) owner and group;

c) **For shared memory and semaphore objects, the creating owner.**

Application Note: The owner and group for some objects, such as TCP/IP sockets and anonymous memory, may be implicit in the process which created the object (and which must have the object mapped for the entire life of the object).

Application Note: The allowed operations for the owner and group for some objects, such as TCP/IP sockets and anonymous memory, may be implicitly set and unchangeable.

Application Note: This requirement includes only implementations where access control attributes are associated with objects rather than subjects. This implementation becomes critical when satisfying FMT_MTD.1.1 and FMT_REV.1.1(1) .

FDP_ACF.1.2 **Refinement**: The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled named objects is allowed:

a) The appropriate access permissions are selected by the following rules:

- **The owning user is compared with the active user. If they match, the access permissions for the owning user are used.**
- **If the owning user does not match the active user, the ACL is examined to determine if there are any ACL entries that match the active user requesting access. If an entry is found that matches (the first one found is used), the access permissions associated with that ACL entry are used.**
- **If no match is found for the active user in the ACL, the owning group is then compared against the active group. If this comparison is successful, the access permissions for the owning group are used.**

- **If the owning group does not match the active group, the ACL is examined to find the first entry (if any) that matches the active group. If such an entry is found, the access permissions associated with that entry are used.**

- **If no match is found for the active group in the ACL, the "others" access permissions are used.**

b) Once a match is found, the permissions are examined to determine if the requested mode of access is one of the allowed access modes for that subject.

Application note: There are three allowed access mode bits in the owner and group permission and in each ACL entry; any combination is syntactically valid (although it may have no semantic meaning). These bits are:

- *read: If this bit is set, the user or group is allowed read access to the object (if allowed by the other access and information flow policies).*

- *write: If this bit is set, the user or group is allowed write access to the object (if allowed by the other access and information flow policies). This also allows append and deletion, though deletion of file system objects depends not on the object's permissions, but on the permissions of the containing directory.*

- *execute: If this bit is set, the user or group is allowed "execute" access to the object (if read access is allowed by the other access and information flow policies). This bit may be ignored or be meaningless for certain types of objects (for example: devices, named First-In First-Out (FIFOs), processes). For directory file system objects, this bit is not interpreted as "execute," but as "search."*

If the particular bit for that type of access is not set, the corresponding user or group is denied that mode of access. Hence, to deny access to a given user or group, the bits for all types of access would not be set.

FDP_ACF.1.3 **Refinement**: The TSF shall explicitly authorise access of subjects to **named** objects based on the following additional rules:

- **The TSF shall allow the overriding of discretionary access controls by an authorised administrator when integrity is set to administrator levels or as operator when particular trusted commands within the TSF are executed at integrity level of at least "operator".**

Application Note: This element allows specifications of additional rules for authorised administrators to bypass the Discretionary Access Control policy for system management or maintenance (e.g., system backup).

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following rules: no additional explicit denial rules.

5.1.2.3 Export of User Data Without Security Attributes (FDP_ETC.1)

FDP_ETC.1.1 The TSF shall enforce the **Mandatory Access Control, Mandatory Integrity Control and Discretionary Access Control policies** when exporting user data, controlled under the SFPs, outside of the TSC.

FDP_ETC.1.2 **Refinement**: The TSF shall export the **unlabeled** user data without the user data's associated security attributes **and shall enforce the following rules when unlabeled user data is exported from the TSC:**

a) Devices used export data without security attributes cannot be used to export data with security attributes unless the change in device state is performed manually and is auditable;

b) The devices used for export do have MAC, MIC, and DAC attributes and these are implicitly the attributes of the exported information;

c) Normal MAC, MIC, and DAC policy rules govern access from subjects to any export device and such access will be auditable;

d) Only an operator or administrator can modify the MAC or MIC attributes of a device and only the owner of a device can modify the device's DAC attributes;

e) All changes to device security attributes are auditable;

f) Devices which are in use by the TSF can not simultaneously be used for export of unlabeled user data.

Application Note: Includes tar files, network devices and cpio files.

5.1.2.4 Export of User Data with Security Attributes (FDP_ETC.2)

FDP_ETC.2.1 **Refinement**: The TSF shall enforce the **Mandatory Access Control, Mandatory Integrity Control, and Discretionary Access Control policies** when exporting **labeled** user data, controlled under the SFPs, outside of the TSC.

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

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

FDP_ETC.2.4 **Refinement:** The TSF shall enforce the following rules when **labeled** user data is exported from the TSC:

a) When data is exported in hardcopy form:

- **The first ("banner") page and last page shall be marked with a printed representation of the "least upper bound" sensitivity label of all data exported to the page;**
- **By default, this marking shall also appear on both the top and bottom of each printed page, though a user whom has been granted a special capability by a user administrator can override this default (and the override is auditable);**
- **An authorised administrator can modify the printed representation of sensitivity labels, though the system provides a sensible default.**

b) If a device is capable of maintaining data security attributes, and the device is being managed by the TSF, the security attributes shall be exported with the data and the device shall completely and unambiguously associate the security attributes with the corresponding data.

c) Devices used to export data with security attributes cannot be used to export data without security attributes unless the change in device state is performed manually and is auditable.

d) The security attributes of exported data shall include Mandatory Access Controls, Mandatory Integrity Controls, and Discretionary Access Controls. This applies to mounted file systems and tape backups.

Application Note: This does not apply to tar files, or cpio. Additionally, data transferred via network will be exported at the level of the process writing to a socket, however, the data itself will possess no MAC or MIC information. Files may possess DAC information (files sent using FTP for instance). Printed hardcopy is the only exported, labeled information that is "human-readable".

5.1.2.5 Complete Information flow control (for Mandatory Access Control Policy) (FDP_IFC.2(1))

FDP_IFC.2.1(1) **Refinement**: The TSF shall enforce the **Mandatory Access Control policy** on **all subjects and all objects**, and all operations that cause that information to flow to and from subjects covered by the **MAC policy**.

FDP_IFC.2.2(1) 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.

5.1.2.6 Complete Information flow control (for Mandatory Integrity Control Policy) (FDP_IFC.2(2))

FDP_IFC.2.1(2) The TSF shall enforce the **Mandatory Integrity Control policy** on **all subjects and all objects**, and all operations that cause that information to flow to and from subjects covered by the SFP.

Application Note: The TOE supports only one subject type: a process (see Appendix B). A discussion of TOE supported objects is in Appendix C. The information for subjects and objects is provided in the appendices so that it is more readily accessible for updating the information.

Application Note: The Mandatory Integrity Control policy is based upon trustworthiness: subjects with a low degree of trustworthiness cannot change data of a higher degree of trustworthiness. A subject with a high degree of trustworthiness cannot be forced to rely on data of a low degree of trustworthiness.

FDP_IFC.2.2(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.

Application Note: The TOE supports only one subject type: a process (see Appendix B). The information for subjects is provided in the appendices so that it is more readily accessible for updating the information.

5.1.2.7 Hierarchical Security Attributes (for Mandatory Access Control) (FDP_IFF.2(1))

FDP_IFF.2.1(1) The TSF shall enforce the **Mandatory Access Control policy** based on the following types of subject and information security attributes:

a) For all subject types (see Appendix B) and information types (see Appendix C): a sensitivity label (current MAC level), consisting of one of at least 16 site definable hierarchical levels and up to a set of 64 site definable non-hierarchical categories;

b) For all subjects: the user's maximum MAC level (which is also a sensitivity label) and the effective user ID;

c) For all information types: the owner ID.

Application Note: The implementation of sensitivity labels does not need to store labels in a format that has the components of the label explicitly instantiated, but may use some form of tag which maps to a level and category set.

Application Note: The sensitivity labels of TCP/IP sockets and anonymous memory objects may be implicit in that it is the same as the subject that created it (and that must always have the object mapped while the object exists).

FDP_IFF.2.2(1) **Refinement**: The TSF shall permit an information flow between a controlled subject and controlled **objects** via a controlled operation if the following rules, based on the ordering relationships between security attributes hold:

a) If the sensitivity label of the subject is greater than (see FDP_IFF.2.7) or equal to the sensitivity label of the object, then the flow of information from the object to the subject is permitted (a read operation);

b) If the sensitivity label of the object is greater than or equal to the sensitivity label of the subject; then the flow of information from the subject to the object is permitted (a write operation);

Application Note: where the label of the object is greater than the label of the subject, this is a blind append (i.e., write does not imply a read).

c) If the information flow is between objects, the sensitivity label of the destination object must be greater than (see FDP_IFF.2.7) or equal to the sensitivity label of the source object.

FDP_IFF.2.3(1) The TSF shall enforce the following information flow control rules:

a) subjects can not operate at mandatory levels above, create objects at mandatory levels above, or change the mandatory level of an object to a level above, the clearance of the owning user;

b) writes to file system and device objects must be at the level of the object (write-up is not allowed);

c) reads to socket device objects must be at the level of the object (read-down is not allowed).

FDP_IFF.2.4(1) The TSF shall provide the following:

a) each user must be assigned a clearance by an authorised administrator where the clearance is within the mandatory range allowed by the system;

b) a user can disable per-page security markings on hardcopy output, if s/he has been granted that capability by an authorised administrator;

c) a user can not upgrade nor downgrade the mandatory level of an object, unless s/he has been granted those capabilities by an authorised administrator.

FDP_IFF2.5(1)The TSF shall explicitly authorise an information flow based on the following rules:

Authorised users may bypass MAC by setting their integrity level to "admin" and executing certain TSF commands. Some commands may also require a user to posses a capability added to their account by the system administrator.

Application Note: See Appendix D for a list of TSF commands, and Appendix E for a list of User commands.

Application Note: These rules regulate the behaviour for each of the roles identified under FMT_SMR.

FDP_IFF.2.6(1) The TSF shall explicitly deny an information flow based on the following rules: no explicit denial rules.

FDP_IFF.2.7(1) **Refinement**: The TSF shall enforce the following relationships for any two valid **sensitivity labels**:

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

- **Sensitivity labels are equal if the hierarchical level of both labels are equal and the non-hierarchical category sets are equal;**
- **Sensitivity label A is greater than sensitivity label B if one of the following conditions exist:**
  - **i) If the hierarchical level of A is greater than the hierarchical level of B, and the non-hierarchical category set of A is equal to the non-hierarchical category set of B.**
  - **ii) If the hierarchical level of A is equal to the hierarchical level of B, and the non-hierarchical category set of A is a proper super-set of the nonhierarchical category set of B.**
  - **iii) If the hierarchical level of A is greater than the hierarchical level of B, and the non-hierarchical category set of A is a proper super-set of the non-hierarchical category set of B.**
- **Sensitivity labels are incomparable if they are not equal and neither label is greater than the other.**

b) There exists a "least upper bound" in the set of **sensitivity labels**, such that, given any two valid **sensitivity labels**, there is a valid **sensitivity label** that is greater than or equal to the two valid **sensitivity labels**;

c) There exists a "greatest lower bound" in the set of **sensitivity labels**, such that, given any two valid **sensitivity labels**, there is a **valid sensitivity label** that is not greater than the two valid **sensitivity labels**.

Application Note: "Sensitivity labels" are equivalent to MAC security attributes.

5.1.2.8 Hierarchical Security Attributes (for Mandatory Integrity Control) (FDP_IFF.2(2))

FDP_IFF.2.1(2) The TSF shall enforce the **Mandatory Integrity Control policy** based on the following types of subject and security information attributes:

a) For all subject types (see Appendix B) and information types (see Appendix C): an integrity label (current MIC level) containing one of at least 8 site-definable hierarchical levels and a subset of a set of at least 16 non-hierarchical categories;

b) For all subjects: the user's maximum MIC level (which is also an integrity label) and the effective user ID;

c) For all information types: the owner ID.

Application Note: Example of such integrity labels are those cited in the Biba Integrity policy.

Application Note: For this family (FDP_IFF) the term "security attributes" refers only to the integrity labels of subject and objects.

FDP_IFF.2.2(2) **Refinement**: The TSF shall permit an information flow between a controlled subject and controlled **objects** via a controlled operation if the following rules, based on the ordering relationships between security attributes hold:

a) If the integrity label of the subject is greater than or equal to the integrity label of the object, then a write (the flow of information from the subject to the object) is permitted;

b) If the integrity label of the object is greater than or equal to the integrity label of the subject; then a read (the flow of information from the object to the subject) is permitted;

c) If the information flow is between objects, the integrity label of the source object must be greater than or equal to the sensitivity label of the destination object.

FDP_IFF.2.3(2) The TSF shall enforce the following information flow control rules:

a) subjects can not operate at integrity levels below, create objects at integrity levels below, or change the integrity level of an object to a level below, the clearance of the owning user;

b) writes to file system and device objects must be at the level of the object (write-down is not allowed);

c) reads to socket device objects must be at the level of the object (read-up is not allowed).

FDP_IFF.2.4(2) The TSF shall provide the following:

a) each user must be assigned a clearance by an authorised administrator where the clearance is within the integrity range allowed by the system;

b) a user can not upgrade nor downgrade the integrity level of an object, unless s/he has been granted those capabilities by an authorised administrator.

FDP_IFF2.5(2)The TSF shall explicitly authorise an information flow based on the following rules:

Authorised users may bypass MIC by setting their integrity level to "admin" and executing certain TSF commands. Some commands may also require a user to posses a capability added to their account by the system administrator.

Application Note: See Appendix D for a list of TSF commands, and Appendix E for a list of User commands.

Application Note: These rules regulate the behaviour for each of the roles identified under FMT_SMR.

FDP_IFF.2.6(2) The TSF shall explicitly deny an information flow based on the following rules: no explicit denial rules.

FDP_IFF.2.7(2) **Refinement:** The TSF shall enforce the following relationships for any two valid **MIC** integrity labels:

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

- **Integrity labels are equal if the hierarchical level of both integrity labels are equal and the non-hierarchical category sets are equal;**
- **Integrity label A is greater than integrity label B if one of the following conditions exist:**

i) If the hierarchical level of A is greater than the hierarchical level of B, and the non-hierarchical category set of A is equal to the non-hierarchical category set of B.

ii) If the hierarchical level of A is equal to the hierarchical level of B, and the non-hierarchical category set of A is a proper super-set of the nonhierarchical category set of B.

iii) If the hierarchical level of A is greater than the hierarchical level of B, and the non-hierarchical category set of A is a proper super-set of the non-hierarchical category set of B.

- **Integrity labels are incomparable if they are not equal and neither label is greater than the other.**

b) There exists a "least upper bound" in the set of integrity labels, such that, given any two valid integrity labels, there is a valid integrity label that is greater than or equal to the two valid integrity labels; and

c) There exists a "greatest lower bound" in the set of integrity labels, such that, given any two valid integrity labels, there is a valid integrity label that is not greater than the two valid integrity labels.

5.1.2.9 Import of User Data without Security Attributes (FDP_ITC.1)

FDP_ITC.1.1 **Refinement**: The TSF shall enforce the **Mandatory Access Control, Mandatory Integrity Control, and Discretionary Access Control policies** when importing **unlabeled** user data, controlled under the SFP, from outside the TSC.

FDP_ITC.1.2 **Refinement:** The TSF shall ignore any security attributes associated with the **unlabeled** user data when imported from outside the TSC.

FDP_ITC.1.3 **Refinement:** The TSF shall enforce the following rules when importing **unlabeled** user data controlled under the SFP from outside the TSC:

a) The TSF shall label the data with the MAC and MIC labels of the device by which the data is imported;

b) When importing data, the data is to be given the effective owner and group attributes of the importer of the data;

c) Devices used to import data without security attributes cannot be used to import data with security attributes unless the change in device state is performed manually and is auditable.

5.1.2.10 Import of User Data with Security Attributes (FDP_ITC.2)

FDP_ITC.2.1 **Refinement:** The TSF shall enforce the **Mandatory Access Control, Mandatory Integrity Control, Discretionary Access Control policies**, when importing **labeled** user data, controlled under the SFP, from outside the TSC.

FDP_ITC.2.2 **Refinement**: The TSF shall use the security attributes associated with the imported **labeled** user data.

FDP_ITC.2.3 **Refinement:** The TSF shall ensure that the protocol used provides for the correct unambiguous association between the **imported** security attributes and the **imported, labeled** user data.

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

FDP_ITC.2.5 **Refinement**: The TSF shall enforce the following rules when importing **labeled** user data controlled under the SFP from outside the TSC:

a) **Devices used to import data with security attributes cannot be used to import data without security attributes unless the change in device state is performed manually and is auditable;**

b) **Sensitivity label, consisting of the following:**

- **A hierarchical level; and**
- **A set of non-hierarchical categories.**

Application Note: Applies only to filesystems and saves on removable media.

5.1.2.11 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.

Application Note: To implement "allocation of the resource to" an object, the TOE may remove the previous content of the resource any time after it was deallocated. Deallocation of a resource may mean that all subjects sharing it have deallocated it. Allocation of a resource to an object may not have meaning until at least one subject accesses the object or brings the object into its address space.

5.1.3 Identification and authentication (FIA)

5.1.3.1 Authentication failure handling (FIA_AFL.1)

FIA_AFL.1.1(1) The TSF shall detect when an authorised administrator configurable positive integer within **1 – 120** of unsuccessful authentication attempts occur related to user **login attempts against any user account on a particular terminal**.

Application Note: The default for the number of unsuccessful authentication attempts is 5 (five). The administrator may also be able ot disable this feature.

FIA_AFL.1.2(1) When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall **lock the terminal by ignoring the SAK**.

Application Note: Additional login attempts are ignored until an administrator unlocks the terminal. Lockouts may not apply to the system console, since doing so might result in an unusable system.

5.1.3.2 Authentication failure handling (FIA_AFL.1(2))

FIA_AFL.1.1(2) The TSF shall detect when an administrator configurable positive integer within **1 - 255** unsuccessful authentication attempts occur related to **login attempts against a particular user account across any set of terminals**.

Application Note: The default may be for this feature to be disabled.

FIA_AFL.1.2(2) When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall **lock the user account**.

Application Note: Additional login attempts on the account are rejected until an administrator unlocks the account. Lockouts may not apply to administrative accounts,since doing so might result in an unusable system.

5.1.3.3 User attribute definition (FIA_ATD.1)

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users:

a) The real user and group identifiers;

b) The clearance of the user (which implicitly specifies which roles the user can operate in);

c) The default Mandatory Access Control (MAC) level;

d) The default Mandatory Integrity Control (MIC) level;

e) Capabilities;

f) The default group;

g) Group membership;

h) Authentication data.

Application Note: Group membership may be expressed in a number of ways: a list per user specifying to which groups the user belongs or a list per group which includes which users are members..

Application Note: A TOE may have two forms of user and group identities, a text form and a numeric form, which have a unique mapping between the representations.

Application Note: The TSF may keep additional security-related data for each user, such as a count of previous authentication failures.

5.1.3.4 Verification of secrets (FIA_SOS.1)

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet:

a) For each attempt to use the authentication mechanism, the probability that a random attempt will succeed is less than one in 300,000,000;

b) For multiple attempts to use the authentication mechanism during a one minute period, the probability that a random attempt during that minute will succeed is less than one in 100,000; and

c) Any feedback given during an attempt to use the authentication mechanism will not reduce the probability below the above metrics.

Application Note: The TOE uses passwords as the secrets. Users are authenticated at login time based on the user name and password they type in at a terminal.

5.1.3.5 User authentication before any action (FIA_UAU.2)

FIA_UAU.2.1 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

5.1.3.6 Single-use authentication mechanisms (FIA_UAU.4)

FIA_UAU.4.1 The TSF shall prevent reuse of authentication data related to **first time boot of a new system, creation of a new password when the password history is enabled, and initial login by a new user**.

5.1.3.7 Re-authenticating (FIA_UAU.6)

FIA_UAU.6.1 The TSF shall re-authenticate the user under the conditions **user attempts to change his own password or user attempts to unlock a screen**.

5.1.3.8 Protected authentication feedback (FIA_UAU.7)

FIA_UAU.7.1 The TSF shall provide only **obscured feedback and notice of authentication failure** to the user while the authentication is in progress.

Application Note: "Obscured feedback" implies the TSF does not produce a visible display of any authentication data entered by a user (such as the echoing of a password), although an obscured indication of progress may be provided (such as an asterisk for each character). It also implies that the TSF does not return any information during the authentication process to the user, which may provide any indication of the authentication data.

5.1.3.9 User identification before any action (FIA_UID.2)

FIA_UID.2.2 The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

5.1.3.10 User-subject binding (FIA_USB.1)

FIA_USB.1.1: The TSF shall associate following user security attributes with subjects acting on behalf of that user:

a) The user identity which is associated with auditable events;

b) The user identity or identities which are used to enforce the Discretionary Access Control Policy;

c) The group membership or memberships used to enforce the Discretionary Access Control Policy;

d) The sensitivity label used to enforce the Mandatory Access Control Policy, which consists of the following:

i) A hierarchical level; and

ii) A set of non-hierarchical categories.

e) The integrity label used to enforce the Mandatory Integrity Control Policy, which consists of the following:

i) A hierarchical level; and

ii) A set of non-hierarchical categories.

f) MAC and MIC clearance;

g) Capabilities.

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users:

a) The sensitivity label associated with a subject shall be within the clearance range of the user;

b) The integrity label associated with a subject shall be within the clearance range of the user;

c) The group associated with the subject shall be one for which the user is currently a member;

d) The user identities which are associated with auditable events and which are used to enforce the Discretionary Access Control Policy shall be the same and shall be that of the user which logged in.

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of a user:

a) Subjects can not change their MAC and MIC levels, but can start other subjects at an upgraded MAC/MIC level (within the user's clearance);

b) Only privileged subjects can change the user ID associated with auditing to a value other than a set-ID program has used in the subject's history (and all ID changes are auditable);

c) The user and group IDs used for DAC policy checks can only be changed by (and after) starting a program which an administrator has specified as a "set-ID" program;

d) Only administrators can change the clearance, capabilities, and group membership of a user;

e) Users can change their default level (within their clearance) and their default group (within their group membership).

5.1.4 Security management (FMT)

5.1.4.1 Management of Security Functions Behaviour (FMT_MOF.1(1))

FMT_MOF.1.1(1) The TSF shall restrict the ability to determine the behaviour of the functions **access change requests for devices** to **the authorised administrators**.

5.1.4.2 Management of Security Functions Behaviour (FMT_MOF.1(2))

FMT_MOF.1.1(2) The TSF shall restrict the ability to modify the behaviour of the function **frequency of audit space threshold warnings (see FAU_STG.4.1)** to the **authorised administrators**.

5.1.4.3 Management of Security Functions Behaviour (FMT_MOF.1(3))

FMT_MOF.1.1(3) The TSF shall restrict the ability to enable the functions **printing hard-copy (see FDP_ETC.2.4)** to **the authorised operators and administrators**.

5.1.4.4 Management of Security Functions Behaviour (FMT_MOF.1(4))

FMT_MOF.1.1(4) The TSF shall restrict the ability to disable or enable the functions **set the "class" of a disk partition (see FDP_ITC.1.3)** to **the authorised operators and administrators**.

5.1.4.5 Management of Security Functions Behaviour (FMT_MOF.1(5))

FMT_MOF.1.1(5) The TSF shall restrict the ability to enable the **functions restore from**

save tape (see FDP_ITC.2.5) to **the authorised operators and administrators**.

5.1.4.6 Management of Security Functions Behaviour (FMT_MOF.1(6))

FMT_MOF.1.1(6) The TSF shall restrict the ability to modify the behaviour of the functions **authentication failure handling (see FIA_AFL.1.1 and 1.2)** to **the authorised administrators**.

5.1.4.7 5.1.4.7 Management of Security Functions Behaviour (FMT_MOF.1(7))

FMT_MOF.1.1(7) The TSF shall restrict the ability to determine the behaviour of and modify the behaviour of the functions **verification of secrets (see FIA_SOS.1.1)** to **the authorised administrators**.

5.1.4.8 Management of Security Functions Behaviour (FMT_MOF.1(8))

FMT_MOF.1.1(8) The TSF shall restrict the ability to disable and enable the functions **mounting and unmounting of file systems (see FDP_ITC.2.5)** to **the authorised administrators and operators**.

5.1.4.9 Management of Security Functions Behaviour (FMT_MOF.1(9))

FMT_MOF.1.1(9) The TSF shall restrict the ability to determine the behaviour of the functions **serial terminal support** to **the authorised administrators**.

5.1.4.10 Management of Security Functions Behaviour (FMT_MOF.1(10))

FMT_MOF.1.1(10) The TSF shall restrict the ability to determine the behaviour of the functions **management of conditions under which abstract machine tests occur, such as during initial start-up, regular intervals, or under specific constraints (see FPT_AMT.1.1)** to **the authorised administrators**.

5.1.4.11 Management of security attributes (for Mandatory Integrity Control) (FMT_MSA.1(1))

FMT_MSA.1.1(1) The TSF shall enforce the **Mandatory Integrity Control policy** to restrict the ability to modify, delete, **and add** the security attributes **set of users with read access rights to the audit records (i.e., users whose clearance includes both maximum security and maximum integrity) (see FAU_SAR.1.1)** to **authorised administrators**.

5.1.4.12 Management of Security Attributes (for Discretionary Access Control) (FMT_MSA.1(2))

FMT_MSA.1.1(2) The TSF shall enforce the **Discretionary Access Control policy** to restrict the ability to modify the security attributes **group membership of users** to **authorised administrators**.

5.1.4.13 Management of security attributes (for Discretionary Access Control) (FMT_MSA.1(3))

FMT_MSA.1.1(3) **Refinement**: The TSF shall enforce the **Discretionary Access Control policy** to restrict the ability to modify the **access control attributes associated with a named object (see FDP_ACF.1.1)** to **authorised administrators and owners of the object**.

5.1.4.14 Management of Security Attributes (for Mandatory Access Control) (FMT_MSA.1(4))

FMT_MSA.1.1(4) The TSF shall enforce the **Mandatory Access Control policy** to restrict the ability to modify the security attributes **value of the sensitivity label associated with an object (see FDP_IFF.2.1(1))** to **authorised administrators or owners with the required capability**.

5.1.4.15 Management of Security Attributes (for Mandatory Integrity Control) (FMT_MSA.1(5))

FMT_MSA.1.1(5) The TSF shall enforce the **Mandatory Integrity Control policy** to restrict the ability to **modify** the security attributes **value of the integrity label associated with an object (see FDP_IFF.2.1)** to authorised **administrators or owners with the required capability**.

5.1.4.16 Management of security attributes (FMT_MSA.1(6))

FMT_MSA.1.1(6) The TSF shall enforce the **Discretionary Access Control policy** to restrict the ability to modify the security attributes **default group for a user** to **authorised administrators**.

5.1.4.17 Management of security attributes (FMT_MSA.1(7))

FMT_MSA.1.1(7) The TSF shall enforce the **Mandatory Access Control policy** to restrict the ability to modify the security attributes **default values for a user's MAC level** to **authorised administrators or that user**.

5.1.4.18 Management of security attributes (for Default Values for Access Control Policies) (FMT_MSA.1(8))

FMT_MSA.1.1(8) The TSF shall enforce the **Mandatory Integrity Control policy** to restrict the ability to change default and modify the security attributes **default values for a user's MIC level (see FMT_MSA.3.1)** to **authorised administrators or that user**.

5.1.4.19 Management of security attributes (FMT_MSA.1(9))

FMT_MSA.1.1(9) The TSF shall enforce the **Mandatory Integrity Control policy** to restrict the ability to change default the security attributes **allowable MIC range of a new file system** to **authorised administrators and operators**.

5.1.4.20 Management of security attributes (FMT_MSA.1(10))

FMT_MSA.1.1(10) The TSF shall enforce the **Mandatory Access Control policy** to restrict the ability to change default the security attributes **allowable MAC range of a new file system** to **authorised administrators and operators**.

5.1.4.21 Management of security attributes (for Roles) (FMT_MSA.1(11))

FMT_MSA.1.1(11) The TSF shall enforce the **Mandatory Integrity Control policy** to restrict the ability to delete **and add** the security attributes **users who are a member of a role (see FMT_SMR.1.1)** to **authorised administrators**.

5.1.4.22 Management of security attributes (for Time) (FMT_MSA.1(12))

FMT_MSA.1.1(12) The TSF shall enforce the **Mandatory Integrity Control policy** to restrict the ability to change default and modify the security attributes **time kept by the TOE (see FPT_STM.1.1)** to **authorised administrators**.

5.1.4.23 Management of security attributes (for TSF-initiated Session Termination) (FMT_MSA.1(13))

FMT_MSA.1.1(13) The TSF shall enforce the **Mandatory Integrity Control policy** to restrict the ability to change default and modify the security attributes **maximum time of user inactivity prior to automatic termination of the session (see FTA_SSL.3.1)** to **authorised administrators**.

5.1.4.24 Secure Security Attributes (FMT_MSA.2)

FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for security attributes.

Application Note: "Secure values" are those defined in the associated guidance documentation. The identity attributes are listed in FDP_IFC.2, and FIA_ATD.1.

5.1.4.25 Static Attributes Initialisation (FMT_MSA.3(1))

FMT_MSA.3.1(1) **Refinement**: The TSF shall enforce the **Discretionary Access Control policy** to provide restrictive default values for security attributes that are used to enforce the **Discretionary Access Control policy**.

FMT_MSA.3.2(1) The TSF shall allow the **authorised administrator** to specify alternative initial values to override the default values when an object or information is created.

5.1.4.26 Static Attributes Initialisation (FMT_MSA.3(2))

FMT_MSA.3.1(2) **Refinement**: The TSF shall enforce the **Mandatory Access Control policy** to provide restrictive default values for security attributes that are used to enforce the **Mandatory Access Control policy**.

FMT_MSA.3.2(2) The TSF shall allow the **authorised administrator** to specify alternative initial values to override the default values when an object or information is created.

5.1.4.27 Static Attributes Initialisation (FMT_MSA.3(3))

FMT_MSA.3.1(3) **Refinement**: The TSF shall enforce the **Mandatory Integrity Control policy** to provide restrictive default values for security attributes that are used to enforce the **Mandatory Integrity Control policy**.

FMT_MSA.3.2(3) The TSF shall allow the **authorised administrator** to specify alternative initial values to override the default values when an object or information is created.

Application Note: The TOE must provide protection by default for all objects at creation time. This may be accomplished through the enforcement of a restrictive default access on objects, or through requiring the user to explicitly specify the desired access controls upon the object at its creation, provided that there is no window of vulnerability through which unauthorised access may be gained to newly-created objects.

5.1.4.28 Management of TSF data (for general TSF data) (FMT_MTD.1(1))

FMT_MTD.1.1(1) The TSF shall restrict the ability to change_default, query, modify, delete, clear, and **create** the **security-relevant TSF data except for audit records, user security attributes, authentication data, and security parameters** to **the authorised administrator**.

Application Note: The restrictions for audit records, user security attributes, authentication data, and critical security parameters are specified below.

5.1.4.29 Management of TSF Data (for audit data) (FMT_MTD.1(2))

FMT_MTD.1.1(2) The TSF shall restrict the ability to query the **audit records** to **authorised administrators**.

5.1.4.30 Management of TSF Data (for audit parameters) (FMT_MTD.1(3))

FMT_MTD.1.1(3) The TSF shall restrict the ability to modify or **observe** the **set of audited events** to **authorised administrators**.

5.1.4.31 Management of TSF Data (for user security attributes) (FMT_MTD.1(4))

FMT_MTD.1.1(4) The TSF shall restrict the ability to **create, delete, and clear** the **audit trail** to **authorised administrators**.

5.1.4.32 Management of TSF Data (for user security attributes, other than authentication data) (FMT_MTD.1(5))

FMT_MTD.1.1(5) The TSF shall restrict the ability to modify and **initialise** the **user security attributes, other than authentication data**, to **authorised administrators**.

5.1.4.33 Management of TSF Data (for authentication data) (FMT_MTD.1(6))

FMT_MTD.1.1(6) The TSF shall restrict the ability to modify the **authentication data** to **authorised administrators and users authorised to modify their own authentication data**.

5.1.4.34 Management of TSF Data (for authentication data) (FMT_MTD.1(7))

FMT_MTD.1.1(7) The TSF shall restrict the ability to **initialise** the **authentication data** to **authorised administrators**.

5.1.4.35 Management of TSF Data (for critical security parameters) (FMT_MTD.1(8))

FMT_MTD.1.1(8) The TSF shall restrict the ability to **create and delete** the **users and groups** to **authorised administrators**.

5.1.4.36 Management of limits on TSF data (FMT_MTD.2)

FMT_MTD.2.1 The TSF shall restrict the specification of the limits for **critical security parameters** to the **authorised administrator or operators**.

FMT_MTD.2.2 The TSF shall take the following actions, if the TSF data are at, or exceed, the indicated limits: **shut down the system**.

Application Note: An example of this would be the amount of audit records exceeding a specified percentage of available hard disk space.

5.1.4.37 Revocation (to authorised administrators) (FMT_REV.1(1))

FMT_REV.1.1(1) The TSF shall restrict the ability to revoke security attributes associated with the users within the TSC to **authorised administrators**.

Application Note: The term "revoke security attributes" means "change attributes so that access is revoked".

FMT_REV.1.2(1) The TSF shall enforce the rules:

a) The revocation of security-relevant authorizations shall be enforced at the next login attempt, trusted command use, or subject creation.

Application Note: Security-relevant authorizations include the ability of authorised users to log in or perform privileged operations. An example of revoking a security-relevant authorization is the deletion of a user account upon which system access is immediately terminated).

5.1.4.38 Revocation (to owners and authorised administrators) (FMT_REV.1(2))

FMT_REV.1.1(2) The TSF shall restrict the ability to revoke security attributes associated with objects within the TSC to **owners and authorised administrators**.

Application Note: The term "revoke security attributes" means "change attributes so that access is revoked".

FMT_REV.1.2(2) The TSF shall enforce the rules:

a) A change in MAC or MIC level of an object will be enforced the next time any subject attempts an access to the object;

b) A change in DAC attributes of an object will be enforced the next time any subject attempts to "open" the object. Subjects which already have the object open will be allowed to continue accessing the object.

c) An owner of an object will only be allowed to change the MAC or MIC level of an object if s/he possesses the appropriate user capability.

Application Note: The state where access checks are made determines when the access control policy enforces revocation. The access control policy may include immediate or delayed revocation. The access rights are considered to be revoked when all subsequent access control decisions made

by the TSF use the new access control information. In cases where a previous access control decision was made to permit an operation, it is not required that every subsequent operation make an explicit access control decision.

5.1.4.39 Time-Limited Authorization (FMT_SAE.1)

FMT_SAE.1.1 The TSF shall restrict the capability to specify an expiration time for **passwords** to **the authorised administrator**.

FMT_SAE.1.2 For each of these security attributes, the TSF shall be able to **lock out the associated authorised user account** after the expiration time for the indicated security attribute has passed.

5.1.4.40 Specification of Management Functions (FMT_SMF.1)

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions:

1) mounting and unmounting of file systems;

2) configure serial lines as login terminals;

3) view and change the threshold for audit space warnings;

4) view and change the password criteria;

5) set or change the password of another user and force another user to change his or her own password;

6) create new users and groups;

7) view the set of users and groups;

8) add and remove users from groups;

9) view, initialize or change the MAC and MIC clearance of a user;

10) view, initialize or change the MAC and MIC ranges of a device;

11) view, set or change the capabilities of a user;

12) view and change the maximum inactivity time on a terminal before locking of the screen or logout of the user's session;

13) display and change the MAC, MIC, and DAC attributes of file system objects owned by other users;

14) save and restore multi-level data to tape;

15) set the class of disk devices;

16) view or modify the auditing parameters (enabled events, users, levels);

17) set the system time;

18) search the audit trail for specific kinds of records;

19) remove audit files;

20) startup the hardcopy printing subsystem;

21) initialize, view, and change the default group, MAC level, and MIC level of any user;

22) create new file systems and specify their MAC and MIC range;

23) display and change the number of allowed authentication failures before a terminal or user account lockout;

24) display and change the length of time of a terminal lockout;

25) configure daemons to automatically run abstract machine self-tests.

5.1.4.41 Security Roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles:

a) authorised administrator;

b) users authorised by the Mandatory Access Control Policy to modify object security attributes;

c) users authorised to modify their own authentication data;

d) operator;

e) user (with no additional authorizations).

Application Note: Any user that is authorised to bypass the MAC, MIC, or DAC policy is, by definition, a trusted user. Operators, as well as administrators, are trusted users, but operators do not have as many authorizations as administrators and their ability to bypass policy is limited by settings controlled by administrators. Otherwise untrusted users may be granted a "capability" by an administrator to modify the MAC and MIC levels of objects s/he owns (within his or her clearance) or may be granted a capability to change their own authentication data. There is no capability to allow a user to modify the DAC attributes of objects which s/he does not own and that function is limited to administrators. The TOE may provide multiple administrator roles (audit administrator, security administrator, etc).

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

5.1.4.42 Assuming Roles (FMT_SMR.3)

FMT_SMR.3.1 The TSF shall require an explicit request to assume the following roles:

a) authorised administrator;

b) Operator;

c) User.

5.1.5 Protection of the TOE Security Functions (FPT)

5.1.5.1 Abstract Machine Testing (FPT_AMT.1)

FPT_AMT.1.1 **Refinement**: The TSF shall run a suite of tests at the request of an **administrator or operator** to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.

Application Note: The TSF may include hardware. In the case that there is no underlying abstract machine upon which the TSF relies to implement required functions, this requirement is met "vacuously" and testing of hardware may fall under FPT_TST.1.

Application Note: The test suite need only cover aspects of the underlying abstract machine on which the TSF relies to implement required functions, including domain separation.

5.1.5.2 Recovery from Failure (FPT_RCV.1)

FPT_RCV.1.1 After **an abrupt power failure or unrecoverable hardware, firmware, or TSF software failure**, the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

Application Note: Other conditions such as non-abrupt power failures (e.g. signaled by a UPS), audit storage runouts, and recoverable TSF failures are handled gracefully, with no need for recovery.

5.1.5.3 Non-Bypassability of the TSF (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.

5.1.5.4 SFP Domain Separation (FPT_SEP.1)

FPT_SEP.1.1 The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2 The TSF shall enforce separation between the security domains of subjects in the TSC.

5.1.5.5 Reliable time stamps (FPT_STM.1)

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

5.1.5.6 TSF Testing (for TSF) (FPT_TST.1)

FPT_TST.1.1 The TSF shall run a suite of self tests during the initial start-up or **at the request of an authorised administrator or operator** to demonstrate the correct operation of the TSF.

FPT_TST.1.2 **Refinement**: The TSF shall provide **administrators and operators** with the capability to verify the integrity of TSF data.

FPT_TST.1.3 **Refinement**: The TSF shall provide **administrators and operators** with the capability to verify the integrity of stored TSF executable code.

5.1.6 TOE Access (FTA)

5.1.6.1 TSF-initiated session locking (FTA_SSL.1)

FTA_SSL.1.1 The TSF shall lock an interactive session after an **administrator configurable time interval of inactivity** by:

a) clearing or overwriting display devices, making the current contents unreadable;

b) disabling any activity of the user's data access/display devices other than unlocking the session.

FTA_SSL.1.2 The TSF shall require the following events to occur prior to unlocking the session: either

a) the user must strike the secure attention key and enter his or her correct password; or

b) an administrator may also be able to unlock the session and terminate it by providing a correct password for the administrator.

5.1.6.2 User-initiated locking (FTA_SSL.2)

FTA_SSL.2.1 The TSF shall allow user-initiated locking of the user's own interactive session by:

a) clearing or overwriting display devices, making the current contents unreadable;

b) disabling any activity of the user's data access/display devices other than unlocking the session.

FTA_SSL.2.2 The TSF shall require the following events to occur prior to unlocking the session: either

a) the user must strike the secure attention key and enter his or her correct password; or

b) an administrator may also be able to unlock the session and terminate it by providing a correct password for the administrator.

5.1.6.3 TSF-Initiated Termination (FTA_SSL.3)

FTA_SSL.3.1 The TSF shall terminate an interactive session after an **administrator configurable time interval of inactivity**.

5.1.6.4 Default TOE Access Banners (FTA_TAB.1)

FTA_TAB.1.1 **Refinement**: Before establishing a user session, the TSF shall display an advisory **notice and consent** warning message regarding unauthorised use of the TOE.

5.1.6.5 TOE Access History (FTA_TAH.1)

FTA_TAH.1.1 **Refinement**: Upon successful session establishment, the TSF shall display the date, time, and location of the last successful session establishment to the session 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.

5.1.6.6 TOE session establishment (FTA_TSE.1)

FTA_TSE.1.1 The TSF shall be able to deny session establishment based on **one of the following conditions**:

a) the terminal is locked due to too many bad authentication attempts being previously detected;

b) the user's account is locked due to too many bad authentication attempts being previously detected;

c) the user's password lifetime has been exceeded;

d) the user's password no longer meets password requirements and the user does not have permission to change the password;

e) minimum mandatory level of the terminal being above the user's clearance;

f) the user's default group is no longer valid and the user does not have permission to change groups;

g) the user is already logged in at another terminal and does not have permission to log in at multiple terminals;

h) the user's account has been locked by an administrator;

i) the user is trying to login to a session at a MAC/MIC level not within the users's clearance;

j) the user's clearance has been revoked from the system.

5.1.7 Trusted path/channels (FTP)

5.1.7.1 Trusted path (FTP_TRP.1)

FTP_TRP.1.1 The TSF shall provide a communication path between itself and local users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from modification or disclosure.

Application Note: This "distinct" path is merely invoked for the duration of its being needed (e.g., for reauthenticating the user); it need not be invoked for the duration of the user's session.

FTP_TRP.1.2 The TSF shall permit local users to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for initial user authentication **and user identification, changing of roles, changing of current group, changing of password, changing of session level, unlocking of a session**.

5.2 End Notes

This section records the functional requirements where deletion of Common Criteria text was performed.

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

   A deletion of CC text was performed in FAU_SAR.1.2. Rationale: The word "user" was deleted to replace it with the defined role of "authorised administrator". This is necessary because only the administrator should be allowed to handle audit records.

   FAU_SAR.1.2 **Refinement**: The TSF shall provide the audit records in a manner suitable for **the authorised administrator** to interpret the information **using a tool to access the audit trail**.

2. FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **[assignment]**

   A refinement of CC text was performed in FDP_ACF.1.2. Rationale: The only type of objects that can have security attributes such as MAC levels assigned to them are named objects. Unnamed objects may be created by processes and have MAC level equivalent to that of the process (subject).

   FDP_ACF.1.2 **Refinement**: The TSF shall enforce the following rules to determine if an operation among subjects and controlled **named** objects is allowed: **[assignment]**

3. FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **[assignment]**

   A refinement of CC text was performed in FDP_ACF.1.3. Rationale: The only type of objects that can have security attributes such as MAC levels assigned to them are named objects. Unnamed objects may be created by processes and have MAC level equivalent to that of the process (subject).

   FDP_ACF.1.3 **Refinement**: The TSF shall explicitly authorise access of subjects to named objects based on the following additional rules: **[assignment]**

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

   A refinement of CC text was performed in FDP_ETF.1.2. Rationale: The LSPP writes the requirement this way to make it clear that data being exported without security attributes has no label and to allow the explicit rules to be specified, as is done for FDP_ETC.2 and FDP_ITC.1.

   FDP_ETC.1.2 **Refinement**: The TSF shall export the **unlabeled** user data without the user data's associated security attributes **and shall enforce the following rules when unlabeled user data is exported from the TSC**:

   a) Devices used export data without security attributes cannot be used to export data with security attributes unless the change in device state is performed manually and is auditable;

   b) The devices used for export do have MAC, MIC, and DAC attributes and these are implicitly the attributes of the exported information;

   c) Normal MAC, MIC, and DAC policy rules govern access from subjects to any export device and such access will be auditable;

   d) Only an operator or administrator can modify the MAC or MIC attributes of a device and only the owner of a device can modify the device's DAC attributes;

   e) All changes to device security attributes are auditable;

   f) Devices which are in use by the TSF can not simultaneously be used for export of unlabeled user data.

5. FDP_ETC.2.1 The TSF shall enforce the [assignment] when exporting user data, controlled under the SFP(s), outside of the TSC.

   A refinement of CC text was performed in FDP_ETC.2.1. Rationale: The LSPP writes the requirement this way to make it clear that data being exported with security attributes has a label.

   FDP_ETC.2.1 **Refinement**: The TSF shall enforce the [assignment] when exporting **labeled** user data, controlled under the SFP(s), outside of the TSC.

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

A refinement of CC text was performed in FDP_ETC.2.2. Rationale: The LSPP writes the requirement this way to make it clear that data being exported with security attributes has a label.

FDP_ETC.2. **Refinement**: The TSF shall export the **labeled** user data with the user data's associated security attributes.

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

   A refinement of CC text was performed in FDP_ETC.2.3. Rationale: The LSPP writes the requirement this way to make it clear that data being exported with security attributes has a label.

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

8. FDP_ETC.2.4 The TSF shall enforce the following rules when user data is exported from the TSC: [assignment].

   A refinement of CC text was performed in FDP_ETC.2.4. Rationale: The LSPP writes the requirement this way to make it clear that data being exported without security attributes has no label and to allow the explicit rules to be specified, as is done for FDP_ETC.2 and FDP_ITC.1.

   FDP_ETC.2.4 **Refinement**: The TSF shall enforce the following rules when **labeled** user data is exported from the TSC: [assignment].

9. FDP_IFC.2.1(1) The TSF shall enforce the [assignment] on [assignment], and all operations that cause that information to flow to and from subjects covered by the SFP.

   A refinement of CC text was performed in FDP_IFC.2.1(1). Rationale: The LSPP changes the wording to make it more clear.

   FDP_IFC.2.1(1) **Refinement**: The TSF shall enforce the [assignment] on [assignment], and all operations that cause that information to flow to and from subjects covered by the **MAC policy**.

10. FDP_IFF.2.2(1) 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: **[assignment**]

    A refinement of CC text was performed in FDP_IFF.2.2(1). Rationale: The words "between a controlled subject and controlled information via a controlled operation if" and "based on the ordering relationships between security attributes hold" was deleted because the information flow for this requirement is among subjects and objects.

    FDP_IFF.2.2(1) **Refinement**: The TSF shall permit an information flow **among subjects and objects** based on the **following rules: [assignment]**

11. FDP_IFF.2.5(1) The TSF shall explicitly authorise an information flow based on the following rules: **[assignment**]

    A refinement of CC text was performed in FDP_IFF.2.5(1). Rationale: Text was refined to tailor it to the TOE. The modified text is a more specific requirement.

FDP_IFF.2.5(1) **Refinement:** Authorised users may bypass MAC by setting their integrity level to "admin" and executing certain TSF commands. Some commands may also require a user to posses a capability added to their account by the system administrator.

12. FDP_IFF.2.7(1) 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.

    A refinement of CC text was performed in FDP_IFF.2.7(1). Rationale: For clarification, "security attribute" was replaced with "sensitivity label". Also the specific ordering rules for this TOE are given, as dictated by the LSPP.

    FDP_IFF.2.7(1) **Refinement**: The TSF shall enforce the following relationships for any two valid **sensitivity labels**:

    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

        1) Sensitivity labels are equal if the hierarchical level of both labels are equal and the non-hierarchical category sets are equal;

        2) Sensitivity label A is greater than sensitivity label B if one of the following conditions exist:

        - **If the hierarchical level of A is greater than the hierarchical level of B, and the non-hierarchical category set of A is equal to the non-hierarchical category set of B.**
        - **If the hierarchical level of A is equal to the hierarchical level of B, and the non-hierarchical category set of A is a proper super-set of the nonhierarchical category set of B.**
        - **If the hierarchical level of A is greater than the hierarchical level of B, and the non-hierarchical category set of A is a proper super-set of the non-hierarchical category set of B.**

        3) Sensitivity labels are incomparable if they are not equal and neither label is greater than the other.

    b) There exists a "least upper bound" in the set of **sensitivity labels**, such that, given any two valid sensitivity labels, there is a valid **sensitivity label** that is greater than or equal to the two valid **sensitivity labels**; and

c) There exists a "greatest lower bound" in the set of **sensitivity labels**, such that, given any two valid sensitivity labels, there is a valid **sensitivity label** that is not greater than the two valid **sensitivity labels**.

13. FDP_IFF.2.2(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: **[assignment**]

    A refinement of CC text was performed in FDP_IFF.2.2(2). Rationale: The words "between a controlled subject and controlled information via a controlled operation if" and "based on the ordering relationships between security attributes hold" was deleted because the information flow for this requirement is among subjects and objects.

    FDP_IFF.2.2(2) **Refinement:** The TSF shall permit an information flow among **subjects and objects** based on the following rules: **[assignment]**

14. FDP_IFF.2.5(2) The TSF shall explicitly authorise an information flow based on the following rules: **[assignment**]

    A refinement of CC text was performed in FDP_IFF.2.5(2). Rationale: Text was refined to tailor it to the TOE. The modified text is a more specific requirement.

    FDP_IFF.2.5(2) **Refinement**: Authorised users may bypass MIC by setting their integrity level to "admin" and executing certain TSF commands. Some commands may also require a user to posses a capability added to their account by the system administrator.

15. FDP_IFF.2.7(2) 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.

    A refinement of CC text was performed in FDP_IFF.2.7(2). Rationale: Requirement is tailored for the TOE.

    FDP_IFF.2.7(2) **Refinement**: The TSF shall enforce the following relationships for any two valid **MIC** 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.

16. FDP_ITC.1 was refined to replace “user data” with “unlabeled user data” in each place those words were used.

    Rationale: The LSPP uses this refinement to clarify that the data being imported had no label.

17. FDP_ITC.2.1 The TSF shall enforce the [assignment] when importing user data, controlled under the SFP, from outside the TSC.

    A refinement of CC text was performed in FDP_ITC.2.1. Rationale: The LSPP uses this refinement to clarify that the data being imported has a label.

    FDP_ITC.2.1 **Refinement**: The TSF shall enforce the [assignment] when importing **labeled** user data, controlled under the SFP, from outside the TSC.

18. FDP_ITC.2.2 **Refinement**: The TSF shall use the security attributes associated with the imported user data.

    A refinement of CC text was performed in FDP_ITC.2.2. Rationale: The LSPP uses this refinement to clarify that the data being imported has a label.

    FDP_ITC.2.2 **Refinement**: The TSF shall use the security attributes associated with the imported **labeled** user data.

19. 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.

    A refinement of CC text was performed in FDP_ITC.2.3. Rationale: This refinement refers to the importation of data from devices such as a mounted floppy drive. The revised wording adds clarity. Also, the LSPP uses this refinement to clarify that the data being imported has a label.

    FDP_ITC.2.3 **Refinement**: The TSF shall ensure that the protocol used provides for the **correct** unambiguous association between the **imported** security attributes and the **imported** user data.

20. 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.

    A refinement of CC text was performed in FDP_ITC.2.4. Rationale: The LSPP uses this refinement to clarify that the data being imported has a label.

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

21. FDP_ITC.2.5 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TSC: [selection].

A refinement of CC text was performed in FDP_ITC.2.5. Rationale: The LSPP uses this refinement to clarify that the data being imported has a label.

FDP_ITC.2.**5 Refinement**: The TSF shall enforce the following rules when importing **labeled** user data controlled under the SFP from outside the TSC: [selection].

22. FMT_MSA.1.1 The TSF shall enforce the [assignment] to restrict the ability to [selection] the security attributes [assignment] to [assignment].

    FMT_MSA.1.1(3) was refined. Rationale: Change wording per LSPP to make the statement more readable.

    FMT_MSA.1.1(3) **Refinement**: The TSF shall enforce the [assignment] to restrict the ability to [selection] the **access control attributes associated with a named object (see FDP_ACF.1.1**) to [assignment].

23. FMT_MSA.3.1 The TSF shall enforce the [assignment] to provide [selection] default values for security attributes that are used to enforce the SFP.

    This requirement was refined in both iterations. Rationale: Change wording per LSPP to make the statement more readable.

    FMT_MSA.3.1(1) **Refinement**: The TSF shall enforce the [assignment] to provide [selection] default values for security attributes that are used to enforce the **Discretionary Access Control policy**.

    FMT_MSA.3.1(2) **Refinement**: The TSF shall enforce the [assignment] to provide [selection] default values for security attributes that are used to enforce the **Mandatory Access Control policy**.

    FMT_MSA.3.1(3) **Refinement**: The TSF shall enforce the [assignment] to provide [selection] default values for security attributes that are used to enforce the **Mandatory Integrity Control policy**.

24. FPT_AMT.1.1 The TSF shall run a suite of tests [**selection**] to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.

A refinement of CC text was performed in FPT_AMT.1.1. Rationale: Administrator implies user with special privileges.

FPT_AMT.1.1 **Refinement**: The TSF shall run a suite of tests at the request of an authorised **administrator** to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.

25. FPT_TST.1.2 The TSF shall provide authorised users with the capability to verify the integrity of [selection].

    A refinement of CC text was performed in FPT_TST.1.2. Rationale: Administrator implies user with special privileges.

    FPT_TST.1.2 **Refinement**: The TSF shall provide authorised **administrators or operators** with the capability to verify the integrity of [selection].

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

    A refinement of CC text was performed in FPT_TST.1.3. Rationale: Administrator implies user with special privileges.

    FPT_TST.1.3 **Refinement**: The TSF shall provide authorised **administrators or operators** with the capability to verify the integrity of stored TSF executable code.

27. FTA_TAB.1.1 Before establishing a user session, the TSF shall display an advisory warning message regarding unauthorised use of the TOE.

    A refinement of CC text was performed in FTA_TAB.1.1. Rationale: Change is a more specific requirement while meeting the original intent.

    FTA_TAB.1.1 **Refinement**: Before establishing a user session, the TSF shall display an **advisory notice and consent** warning message regarding unauthorised use of the TOE.

28. FTA_TAH.1.1 Upon successful session establishment, the TSF shall display the [selection] of the last successful session establishment to the user.

    A refinement of CC text was performed in FTA_TAH.1.1. Rationale: Change gives more specific and relevant detail.

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

5.3 TOE Security Assurance Requirements

All TOE security assurance requirements are drawn from the CC Part 3. The combination of chosen assurance components result in an Evaluated Assurance Level 5, augmented (EAL5+). As shown, ALC_FLR.3 and ATE_IND.3 go beyond the requirements for EAL5. The intended TOE environment and the value of information processed by this environment establish the need for the TOE to be evaluated at this EAL level. These security assurance requirements are summarised in *Table 3: Assurance Requirements – EAL5 Augmented.*

Table 3: Assurance Requirements – EAL5 Augmented

Assurance Class	Assurance Family	Assurance Components by Evaluation Assurance Level						
		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Configuration	ACM_AUT					1		
Management	ACM_CAP					4		
	ACM_SCP					3		
Delivery and	ADO_DEL					2		
Operation	ADO_IGS					1		
Development	ADV_FSP					3		
	ADV_HLD					3		
	ADV_IMP					2		

Assurance Class	Assurance Family	Assurance Components by Evaluation Assurance Level						
		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
	ADV_INT					1		
	ADV_LLD					1		
	ADV_RCR					2		
	ADV_SPM					3		
Guidance	AGD_ADM					1		
Documents	AGD_USR					1		
Life Cycle	ALC_DVS					1		
Support	ALC_FLR							
	ALC_LCD					2		
	ALC_TAT					2		
Tests	ATE_COV					2		
	ATE_DPT					2		
	ATE_FUN					1		
	ATE_IND				2			3
Vulnerability	AVA_CCA					1		
Assessment	AVA_MSU				2	2		
	AVA_SOF				1	1		
	AVA_VLA				2	3		

5.4 Strength of TOE Security Functional Requirements

This ST includes AVA_SOF.1. This TOE is intended to meet a strength of SOF-high. SOF applies to FIA_SOS.1. The strength of the “secrets” mechanism is consistent with the objectives of authenticating users (O.User_Authentication), operators, and administrators and with maintaining a separate administrator role (O.Admin_Role).

5.5 Security Requirements for the IT Environment

XTS-400 version 6.4(UKE) places no security requirements on the IT environment.

This page is intentionally left blank

Section 6 TOE Summary Specification

6.1 Measures Used to Meet IT Security Functions

The following functions implement the SFRs presented in Section *5.1*.

Table 4: Security Functions and the SFRs They Implement

Security Function	**SFRs**
AUDGEN	FAU_GEN.1, FAU_GEN.2, FAU_STG.2, FAU_STG.3, and FAU_STG.4 and contributes to FMT_MTD.2
AUDREV	FAU_SAR.1, FAU_SAR.2, and FAU_SAR.3
DACENF	FDP_ACC.2 and FDP_ACF.1 and contributes to FDP_ETC.1, FDP_ETC.2, FDP_ITC.1, FDP_ITC.2, FIA_USB.1, and FMT_REV.1
MACENF	FDP_IFC.2(1), FDP_IFF.2 (1), FIA_USB.1and contributes to the implementation of FAU_SAR.2, FDP_ETC.1, FDP_ETC.2, FDP_ITC.1, FDP_ITC.2 and FMT_REV.1
MICENF	FDP_IFC.2(2), FDP_IFF. (2), and FIA_USB.1, and contributes to the implementation of FAU_STG.2, FDP_ETC.1, FDP_ETC.2, FDP_ITC.1, FDP_ITC.2, and FMT_REV.1
RIPENF	FDP_RIP.2
IDNAUT	FAU_SAA.1, FIA_AFL.1, FIA_ATD.1, FIA_SOS.1, FIA_UAU.2, FIA_UAU.4, FIA_UAU.7, FIA_UID.2, FTA_SSL.1, FTA_SSL.2, FTA_SSL.3, FTA_TAB.1, FTA_TAH.1 and FTA_TSE.1 contributes to FMT_SAE.1 and FIA_USB.1
SECMGT	FMT-MOF.1, FMT_MSA.1, FMT_MSA.2, FMT_MSA.3, FMT_MTD.1, FMT_MTD.2, FMT_REV.1(1), FMT_REV.1(2), FMT_SAE.1, FMT_SMF.1, FAU_SEL.1, FIA_USB, FMT_SMR.1, FMT_SMR.3
SECFPR	FPT_AMT.1, FPT_RCV.1, FPT_RVM.1, FPT_SEP.1, FPT_STM.1, and FPT_TST.1
TRUPTH	FPT_TRP.1

Table 5: SFRs and the Security Functions that Satisfy Them

Class Name	Functional Family SFR (Component)	Security Functions	Rationale that Security Function is Suitable to Meet SFR
Security Audit	FAU_GEN.1	AUDGEN	Includes the mechanism for generating audit records
	FAU_GEN.2	AUDGEN	Places the real user ID in each audit record
	FAU_SAA.1	IDNAUT	Monitors the number of failed login attempts and takes action when a threshold is reached
	FAU_SAR.1	AUDREV	Provides a trusted **audit** command that formats audit records for human-readable display and allows an administrator to view records in the audit trail
	FAU_SAR.2	AUDREV, MACENF	AUDREV restricts audit record display to administrators executing at maximum security. MACENF prevents also prevents any user not cleared to system maximum from reading the audit trail in a raw fashion, as the audit files are at maximum security.
	FAU_SAR.3	AUDREV	Allows the administrator to search the audit trail for specific kinds of records
	FAU_SEL.1	SECMGT	Provides a trusted **param_edit** command which allows pre-selection of auditing by event type, user, and MAC/MIC level
	FAU_STG.2	MICENF, AUDGEN	MICENF is used to prevent the modification or deletion of records in the audit trail by untrusted users, as the audit files and the directory in which they reside are at maximum integrity. AUDGEN includes a mechanism by which sufficient disk space is held in reserve such that if additional permanent storage for audit records runs out, all buffered records can be written to permanent storage. In this case, the operator is notified of the problem and the system is shut down.
	FAU_STG.3	AUDGEN	Includes monitoring of the number of audit files and if an administratively set threshold is exceeded, the operator and/or administrator is notified via a console message.
	FAU_STG.4	AUDGEN	Includes detection of runout of permanent storage (disk) space for the audit trail. When this is detected, the system is shut down to prevent generation of too much additional audit data. Disk space is held in reserve so that if this happens, all buffered records can be written to permanent storage.

Class Name	Functional Family SFR (Component)	Security Functions	Rationale that Security Function is Suitable to Meet SFR
User Data Protection	FDP_ACC.2	DACENF	All subjects have a user and group ID and all objects have either an explicit or implicit set of discretionary access attributes. DAC checks are always performed, unless the subject created the object or inherited the object from its parent.
	FDP_ACF.1	DACENF	Includes the specific attributes needed for DAC and the specific rules used for DAC.
	FDP_ETC.1	DACENF, MACENF, MICENF	Data exported without security attributes will implicitly have the attributes of the device (user-class disk, non-save tape, network) used to export the data. The security functions will ensure that the subject has the appropriate access to the export device.
	FDP_ETC.2	DACENF, MACENF, MICENF	When data is exported with security attributes, the security attributes are associated with the data in different ways depending on the export media. MIC and DAC attributes are not exported to hard-copy output, but MAC attributes are written on the banner page and the header and footer of every other page. For save tapes, all attributes are bound directly with each data object and maintained transparently by the tape media as part of the data. For disk file systems, all attributes are kept in a branch which is found via object ID and which in turn identifies the disk blocks holding the object data.
	FDP_IFC.2	MACENF, MICENF	MACENF defines the mandatory access control policy. MICENF defines the mandatory integrity control policy.
	FDP_IFF.2	MACENF, MICENF	MACENF includes the specific attributes needed for MAC and the specific rules used for MAC. MICENF includes the specific attributes needed for MIC and the specific rules used for MIC.
	FDP_ITC.1	DACENF, MACENF, MICENF	Data imported without security attributes will implicitly have the attributes of the device (user-class disk, non-save tape, network) used to import the data. The security functions will ensure that the subject has the appropriate access to the import device.
	FDP_ITC.2	DACENF, MACENF, MICENF	When data is imported with security attributes, the security attributes are associated with the data in different ways depending on the import media. For save tapes, all attributes are extracted directly from each data object. For disk file systems, all attributes are obtained from the branches associated with the objects.

Class Name	Functional Family SFR (Component)	Security Functions	Rationale that Security Function is Suitable to Meet SFR
	FDP_RIP.2	RIPENF	Provides all mechanisms to avoid residual data problems.
Identification and Authentication	FIA_AFL.1	IDNAUT	Includes a mechanism to lock the terminal if too many failed login attempts are detected.
	FIA_ATD.1	IDNAUT	Defines the security attributes maintained for users.
	FIA_SOS.1	IDNAUT	Implements a password authentication mechanism. This includes defining the strength of the mechanism and defining various restrictions on password selection by users.
	FIA_UAU.2	IDNAUT	The system can not be used until a user successfully logs in and no successful log in is allowed, unless the user provides the correct password for the user name entered.
	FIA_UAU.4	IDNAUT	The initial passwords provided in a new system are marked as expired, so that they will have to be changed during the initial login. When administrators create new user accounts, they can specify that the initial password can only be used once.
	FIA_UAU.7	IDNAUT	The mechanism will not echo to the terminal, the password which is entered by the user.
	FIA_UID.2	IDNAUT	The system can not be used until a user successfully logs in and no successful log in is allowed, unless the user provides a valid user name.
	FIA_USB.1	DACENF, IDNAUT, MACENF, MICENF, SECMGT	IDNAUT includes starting a user's session at the default MAC and MIC levels, and with the default group, for that user. DACENF includes checking that the default group is within the group membership of the user. MACENF and MICENF include checking that the default MAC and MIC levels, respectively, are within the user's clearance and within the allowed ranges for the terminal and system. SECMGT includes allowing a user to change his or her session's MAC, MIC, and group and includes allowing an administrator to change a user's clearance, group membership, and capabilities.

Class Name	Functional Family SFR (Component)	Security Functions	Rationale that Security Function is Suitable to Meet SFR
Security Management	FMT_MOF.1	SECMGT	Includes trusted commands restricted to the administrator which allow: 1. setting of a threshold for the number of new audit files before the operator is warned; 2. setting of the allowable MAC and MIC ranges of devices; 3. configuring of a daemon to perform abstract machine and TSF self-tests at designated times; 4. setting of the threshold for the number of allowed, failed login attempts; 5. setting of the lockout interval on a terminal once the threshold of failed login attempts has been hit; 6. setting of parameters for valid passwords; 7. configuring a serial device as a login terminal. Restricts to administrators and operators the trusted commands which provide the following functions: 1. startup of hard-copy printing facilities; 2. setting the "class" of a disk device to "user" or "trusted"; 3. restoring multi-level data from save tape; 4. mounting and unmounting of file systems.
	FMT_MSA.1	SECMGT	Includes trusted commands restricted to the administrator which allow: 1. setting of user clearances, such that a very limited set will be allowed to act as administrators or operators; 2. change the group membership of users; 3. setting of the MAC, MIC, and DAC attributes of objects not owned by the administrator, including restrictions such that access by some subjects would be revoked; 4. change of the default MIC and MAC level and the default group for another user; 5. setting the system time; 6. setting the terminal inactivity time (after which a session will be logged out). Includes trusted commands restricted to the administrator and operator which allow: 1. setting initial MAC and MIC range for a new file system. Includes trusted commands available to untrusted users which allow: 1. setting of the MAC, MIC, and DAC attributes of objects owned by the user (though MAC and MIC changes require the appropriate capability), including restrictions such that access by some subjects would be revoked; 2. changing the default MIC and MAC level of the user.
	FMT_MSA.2	SECMGT	The trusted commands that provide for "management" check that input security attributes are within allowable limits.

Class Name	Functional Family SFR (Component)	Security Functions	Rationale that Security Function is Suitable to Meet SFR
	FMT_MSA.3	SECMGT	Includes trusted commands that allow a user to set his or her default MAC level, MIC level, and group. Includes a trusted command which allows an administrator to set the default MAC level, MIC level, and group of any user.
	FMT_MTD.1	SECMGT	Restricts to administrators the trusted commands which allow creation, modification and deletion of all security data, except: 1. a user's own password; 2. a user's own default MAC, MIC, and group; 3. a device's current level. Restricts to administrators the trusted commands which allow viewing of the following security data: 1. audit records; 2. enabled/disabled audit events, users, and levels. Restricts to administrators and operators the trusted commands which allow modification of the following security data: 1. a device's current MAC and MIC level. Includes trusted commands available to untrusted users which allow: 1. changing the user's password (if the user has the required capability).
	FMT_MTD.2	AUDGEN, SECMGT	SECMGT includes trusted operator and administrator commands to specify the size of a file system and a threshold number of audit files. AUDGEN includes a mechanism that checks this threshold and generates a console message if the threshold is reached. AUDGEN also includes a mechanism to detect complete runout of disk space for the audit trail, at which time the system will be shut down.
	FMT_REV.1	DACENF, MACENF, MICENF, SECMGT	SECMGT includes the only TSF interface to allow the security attributes of a user to be changed and this interface is restricted to administrators. SECMGT includes the only TSF interface to allow one user to modify the security attributes of an object s/he does not own and this interface is restricted to administrators. SECMGT, DACENF, MACENF, and MICENF include the only TSF interfaces to allow a user to modify the security attributes of an object s/he does own. DACENF, MACENF, and MICENF include the revocation access checks.
	FMT_SAE.1	IDNAUT, SECMGT	SECMGT includes an interface to allow an expiration time to be set on passwords and this interface is restricted to administrators. IDNAUT checks the expiration time during a login attempt and prevents the login if the password to too old.
	FMT_SMF.1	SECMGT	SECMGT is defined by the list given.
	FMT_SMR.1	SECMGT	Includes a mechanism to support different roles.

Class Name	Functional Family SFR (Component)	Security Functions	Rationale that Security Function is Suitable to Meet SFR
	FMT_SMR.3	SECMGT	Different roles can only be established by the explicit action, on the part of the user, of either logging in or changing the current session MIC level. In the later case the new MIC level must be within the clearance level of the user.
Protection of the TOE Security Functions	FPT_AMT.1	SECFPR	Includes the abstract machine test mechanism. Note that this mechanism may actually be part of the TSF self-test mechanism.
	FPT_RCV.1	SECFPR	Includes a non-automatic recovery mechanism using specific trusted commands that execute while the TOE is in a maintenance mode.
	FPT_RVM.1	SECFPR	Includes the mechanisms, which depend on object type and specific policy, to implement a reference validation monitor.
	FPT_SEP.1	SECFPR	Includes the mechanisms, which depend on the kinds of interference being prevented and the portion of the TSF being protected, to implement separation of the TSF and non-TSF domains. Also includes the mechanism to separate subject domains from one another.
	FPT_STM.1	SECFPR	Includes the mechanism for providing time stamps.
	FPT_TST.1	SECFPR	Includes the TSF self-test mechanism.
TOE Access	FTA_SSL.1	IDNAUT	Includes an inactivity timeout mechanism whereby the user's session will be locked if it is inactive for longer than an administratively set threshold.
	FTA_SSL.2	IDNAUT	Includes a mechanism whereby the user's session will be locked if explicitly requested by the user.
	FTA_SSL.3	IDNAUT	Includes an inactivity timeout mechanism whereby the user's session will be logged out if it is inactive for longer than an administratively set threshold.
	FTA_TAB.1	IDNAUT	Includes login functionality that displays an administratively set of notices and warnings.
	FTA_TAH.1	IDNAUT	Includes login functionality that, after a successful login, notifies a user of information about the previous successful and unsuccessful logins.
	FTA_TSE.1	IDNAUT	Includes login functionality that can deny the login under certain conditions.
Trusted Path/Channels	FTP_TRP.1	TRUPTH	Defines the trusted path mechanism.

6.1.1 Audit Generation – AUDGEN

The security audit generation function, AUDGEN, is a distributed function that generates audit records for the security related events identified FAU_GEN.1 and Appendix F. Audit events are generated by Trusted Software, Operating System Software, TSF System Services, and the Kernel and include the following types of events:

- Startup and shutdown of the operating system
- Use of special permissions that circumvent the access control policies
- Login attempts
- Logout commands issued
- Opens and closes of file system objects
- Creates and deletes of file system objects
- Operator commands issued
- Administrator commands issued
- Print request issued with no markings.

Each audit record has a header that contains the size of the audit record, type of event being audited, date and time of audit record generation, process ID of the process causing the audit event, MAC and MIC label of the process, effective privileges of the process, real user ID, and real group ID.

When a threshold number of recent audit files have been generated, a message is sent for the administrator and operator at the system console (and all console messages can be optionally written to a log). Audit files consume space on the boot file system, which can fill up due to the audit trail or to other system activity, and are not placed on any other disk. The warning message to the administrator gives him or her a chance to remove old data from the boot file system or to archive old audit files on tape.

Disk space is held in reserve by the audit mechanism so that if disk space for permanent storage of the audit trail runs out, all buffered records can still be written. When this happens, the operator is notified of the problem and the system is shut down due to the inability to continue generating audit data.

The administrator can set an “audit sync interval” in order to minimize the amount of audit data that could be lost during a system crash.

The audit trail is a set of audit files, in a special directory, that are collections of all the audit records generated by request from various parts of the TSF. The audit files and directory reside at administrator integrity so that only an administrator could modify or remove them. The audit files reside at maximum security and are protected by a TSF-internal subtype mechanism so that untrusted users can not read them.

When the administrator changes the set of audit events, users, or levels to be audited (which is the only way auditing can be enabled or disabled), the changes do not take effect until the system is rebooted. Auditing is always started up during system booting, assuming some events are enabled –

no explicit action by a trusted user is necessary. The only exception is that if the boot file system is out of space during boot, auditing will not be started and a message to that effect will be presented to the operator. Note that auditing is not enabled during boots from CD because the media is not writeable.

6.1.2 Audit Review – AUDREV

The security audit review function, AUDREV, provides the authorised administrator the tools necessary to examine and review the audit records generated by the AUDGEN function.

The Trusted Software audit command accepted option of “display” is used to view the audit data through the use of three subcommands (print, reset, and select). It formats the raw audit data to allow management of the audit files via one of five options (switch, remove, files, display, and exit). No other trusted command is provided for reviewing the audit trail. Because the audit files are at maximum integrity and protected by a TSF-internal subtype mechanism, there is no other way in an evaluated configuration to review the audit.

Operators can use the audit command, but can not use it to review the audit trail. They can only see which audit files exist and request that a new file be started.

Valid audit selection criteria are audit event type, start date and time, stop date and time, process ID, user ID, group ID, device ID, trusted editor command name, trusted editor request, file name, object level, and subject level.

6.1.3 Discretionary Access Control – DACENF

The discretionary access control function, DACENF, enforces the Discretionary Access Control policy for the STOP OS, based on user identity and group membership of the subject and owner identity, group, and operation allowed for the object.

The TOE allows owning users to define and control access to named objects through the use of an Access Control List (ACL). Every subject has associated with it an effective user and group; every named object has an ACL. Each ACL contains permissions that specify the allowable access for the owning user, the owning group, up to seven other user or groups, and any user or group not explicitly listed. These permissions can either grant or deny a particular form of access to a named object. When a subject introduces an object into its address space, the ACL is checked to ensure that the subject can access the object.

The kinds of access that are controlled are read, write, and execute. Write does not imply the ability to delete and some objects can not be executed.

Only administrators can introduce new users and groups to the system, establish the group membership of users, or set the default group for users. Normal users can change the discretionary attributes of only the objects they own, but administrators can change the attributes of any object.

6.1.4 Mandatory Access Control – MACENF

The mandatory access control function, MACENF, enforces the Mandatory Access Control policy for the STOP OS, based on level and categories of the subject and classification (level and category(ies)) of the object.

The TSF enforces a mandatory access control policy over all identified system resources (i.e., subjects, storage objects, and I/O devices) that are accessible, either directly or indirectly, to subjects external to the TSF. As the basis of its enforcement, this policy uses MAC labels that are associated with every subject and object in the system. These MAC labels consist of 16 hierarchical sensitivity levels, and 64 nonhierarchical sensitivity categories.

The TOE provides a dominates function that is used to compare sensitivity labels; this comparison is done whenever a subject external to the TSF accesses an object. Every user has an identification and authentication database record that specified the MAC label of the user's clearance. The TSF enforces the restriction that any subject created on behalf of a user has a current MAC label dominated by the user's clearance.

The kinds of access that are relevant are read and write – execute is considered the same as read. The MAC level of processes and some objects can not be modified. Only administrators can change the MAC level of an object, except that a user (who has been granted an appropriate capability) can change the level of objects that s/he owns. A MAC level change to an object will take effect immediately, even if that means denying access to the object by a process which already has the object "open".

Mandatory security control is used internally by the TSF to prevent viewing of sensitive TSF data, including the audit trail and authentication data.

The TSF is designed to minimize the covert channels that exist to circumvent the MAC policy. Potential use of covert storage channels is detected and a resource exhaustion delay is used to reduce covert storage channel capacity without eliminating any user capabilities. The delay is imposed only when a resource exhaustion error occurs. The resource exhaustion delay is a site-set time interval that puts processes to sleep for a set amount of time when a resource exhaustion condition is encountered. The param_edit command is used to change the default value. Potential use of a storage channel is also an auditable event.

6.1.5 Mandatory Integrity Control – MICENF

The mandatory integrity control function, MICENF, enforces the Mandatory Integrity Control policy for the STOP OS, based on integrity level of the subject and integrity level of the object.

The TSF enforces a mandatory integrity control policy over all identified system resources (i.e., subjects, storage objects, and I/O devices) that are accessible, either directly or indirectly, to subjects external to the TSF. As the basis of its enforcement, this policy uses MIC labels that are associated with every subject and object in the system. These MIC labels consist of 8 hierarchical integrity levels, and 16 nonhierarchical integrity categories.

The TOE provides a *dominates* function that is used to compare integrity labels; this comparison is done whenever a subject external to the TSF accesses an object. Every user has an identification and authentication database record that specified the MIC label of the user's clearance. The TSF

enforces the restriction that any subject created on behalf of a user has a current MIC label that dominates the user's MIC clearance.

The kinds of access that are relevant are read and write – execute is considered the same as read. The MIC level of processes and some objects can not be modified. Only administrators can change the MIC level of an object, except that a user (who has been granted an appropriate capability) can change the level of objects that s/he owns. A MIC level change to an object will take effect immediately, even if that means denying access to the object by a process which already has the object "open".

Mandatory integrity control is used internally by the TSF to prevent modification or deletion of TSF data, including the audit trail and configuration parameters for "alarm" mechanisms (such as low disk space, low audit trail space, excessive failed login attempts).

6.1.6 Residual Information Protection – RIPENF

The residual information protection function, RIPENF, provides object reuse control for the STOP OS. The TSF is designed to prevent a process from seeing any information left over in an object or memory area from use by another process (or the TSF itself). Generally, main memory and device memory/media/registers are cleared as needed, in a manner dependent on the resource in question.

When a file is deleted, the name of the file is deleted, leaving no residual information in the directory (this is a variation from the standard UNIX behaviour). When a segment shrinks, the TSF clears residual data in partial pages.

6.1.7 Identification and Authentication – IDNAUT

The identification and authentication function, IDNAUT, provides and controls the identification and authentication of users of the STOP OS. Users are required to enter their user name and password prior to gaining access to any other functions of the system. User names are converted to a user-id. Passwords are encrypted and compared to the encrypted password, for that user-id, in the user authentication database. Passwords are never stored in the clear and the encryption mechanism used is relatively strong encryption (128-bit MD5). In addition the passwords are stored in a database that is protected with a TSF-internal subtype so that not even users cleared for maximum security can read it.

The administrator can specify the following system-wide parameters for passwords:

- minimum password length
- mixed case required
- non-alphanumeric characters required
- password history size (if the history is of size N, when a user changes his or her password, it can not be the same as any of his or her last N passwords)
- expiration time (after which the user must change the password)
- lifetime (after which the user can not log in nor change the password)

- maximum consecutive erroneous passwords
- not to be found in a dictionary.

The identification and authentication function can deny login attempts due to issues with the user's password (as shown above), clearance, group, or ability to log in multiple times.

The identification and authentication function, IDNAUT, also provides advisory banners prior to user identification and authentication and login history after successful identification and authentication. The login history includes the number of failed login attempts since the last logout, potentially alerting the user that someone has been trying to break into his or her account.

The identification and authentication function, IDNAUT, also associates attributes with a user following a successful login. These attributes are set by an administrator, except that the user can change the default MAC and MIC levels, within his or her administratively set clearance.

The identification and authentication function, IDNAUT, also provides TSF-initiated locking or termination of interactive user sessions after an administratively defined inactivity timeout occurs. This serves primarily to lessen the chance that a user "walks away" from his session such that another person can make unauthorised use of the session, but serves partly as a screen saver. The user can also explicitly lock his or her session. Authentication is required to unlock a session.

The identification and authentication function, IDNAUT, also monitors the number of failed login or session unlock attempts. When an administratively configured threshold is reached, a console message is generated for the administrator and either the user account or the terminal may be locked for an administratively set time. Only an administrator can unlock a user account or terminal.

6.1.8 Security Management – SECMGT

The security management function, SECMGT, provides the management functions to support most of the other functions. It allows authorised administrators and, in some cases operators, to specify initial values, change values, and monitor (as appropriate) the security parameters that affect the behaviour of the TOE.

SECMGT also provides for three distinct user roles (administrator, operator, and user). The roles are implemented using MIC levels for users. Normal users have a clearance that specifies a maximum integrity level below that required for the operator role. Operators execute at an integrity level below that required by the administrator role.

All TSF data, other than the runtime data dynamically held by the TSF, resides in files that are at the administrator integrity level. This prevents any user other than an administrator, or any process running below administrator integrity, from modifying the data. Some trusted commands have privileges that allow operators to modify some of these data items in controlled manners.

A normal user is not allowed to modify or delete TSF data, but is allowed to change some of the security attributes of their own login session or their own objects.

Use of all commands restricted to operators and administrators is auditable. This includes commands that would be used to archive and restore audit files, and that would be used for multi-level backups and restores in general (and the opening and closing of devices on which to archive/restore, and of the files themselves, are also auditable).

6.1.9 Security Function Protection – SECFPR

The security function protection function, SECFPR, provides protection mechanisms for the security functions of the TOE. The TSF is carefully designed to protect itself from tampering or bypass by untrusted code and users. This includes mechanisms to:

- test the abstract machine to ensure that features of the hardware and firmware that are relied upon for secure operation are functioning properly;
- self-test the TSF itself to verify that it has not been tampered with or affected by a bug;
- avoid tampering of the TSF by untrusted code;
- avoid tampering of one process by another;
- avoid bypass of the access control mechanisms by untrusted code;
- repair file systems in a secure fashion after a system crash.

The abstract machine tests and TSF self-tests are provided together in a single trusted command. The operator, or any user, can run these whenever desired and the administrator can configure them to run periodically in the background. Normal users, however, are not allowed to generate new checksum information for the TSF files, since that could impede the ability of trusted users to correctly make use of the checksums. As the hardware is included in the TSF, the abstract machine tests could really be considered part of the TSF self-tests.

In the event that the TOE crashes, the file systems that were mounted at the time of the crash will have to be repaired before they can be used again. The TSF will prevent booting from, or mounting, a file system that was not previously properly unmount or repaired, since the security attributes for objects in the file system may be corrupt (in rare circumstances). Two trusted commands are provided to check the validity of file systems and to repair them. They are usable only by operators and administrators. The boot file system is repaired in a maintenance mode.

6.1.9.1 How Domain Separation is Provided by the TOE

The maintenance of a security domain for the TSF's execution that protects it from interference and tampering by untrusted subjects (FPT_SEP.1.1) is done partly by hardware and partly by software. The hardware provides primitive building blocks to support this as follows:

- Has a privilege domain mechanism that allows software executing in a privileged domain to be inaccessible to software executing in an unprivileged domain and that allows privileged software to restrict access to I/O instructions to privileged software. There are at least two privilege levels.
- Hardware descriptors control which domains can access which areas in physical memory and control where unprivileged software can transfer into privileged software. These descriptors are only modifiable by hardware-privileged software.
- Unprivileged software has no way to disable or circumvent the descriptors and has no way to gain privilege, other than by making controlled transfers to privileged software through the aforementioned descriptors.

- Unprivileged software has no way to modify other hardware descriptors, the mode of operation of the hardware, nor the control and configuration registers or the hardware.

However, the hardware does not “know”:

a) which software is trusted and which is not;

b) where the allowable entry points into the TSF are.

Also, some aspects of providing the domain separation are implemented completely in software, with no reliance on the hardware building blocks. The TSF (software) implements the protection, with full awareness of the security state and requirements, by:

a) marking and distinguishing code as trusted or untrusted by virtue of integrity and privilege attributes attached to the containing file system object;

b) preventing creation or modification, by untrusted code, of file system objects containing trusted code;

c) starting untrusted software in an unprivileged domain and requiring TSF involvement in creation of new subjects;

d) avoiding use of or transfers to untrusted code (fragments) from within privileged domains;

e) controlling exactly where and how untrusted code can make requests of the TSF;

f) providing a very limited, well-defined, and secure set of services that unprivileged code can request;

g) checking the location of parameters specified by unprivileged code to ensure that they are not within the privileged domain(s);

h) setting hardware flags to a safe state for the TSF upon entry to the TSF from untrusted code (where some of these flags are settable by untrusted code);

i) implementing (fully in software) domain attributes for semaphore, shared memory, and process objects and using those attributes to prevent the TSF’s internal objects from being created or manipulated by unprivileged code;

j) using the mandatory integrity mechanism to prevent creation or modification of TSF objects by untrusted code;

k) setting up all hardware descriptors for TSF-internal objects, processes, and mechanisms such that the hardware will cause a fault if unprivileged code attempts to use the descriptor;

l) setting hardware flags to make I/O instructions cause a fault if attempted by unprivileged code;

m) setting up all hardware fault and trap descriptors so that the TSF will catch all interrupts and faults from the hardware;

n) refusing to perform any service on behalf of untrusted code, or to allow the attempted use to proceed, when the hardware generates a fault due to the untrusted code attempting to use a descriptor, register, or instruction that is restricted to privileged code;

o) refusing to allow unprivileged code to hog the CPU or to fill up the audit trail to the point that the TSF security functions would become disabled or malfunction [note that unprivileged code may still be able to slow down TSF operations significantly or may precipitate an orderly shut down of the system];

p) setting the CPU state to enable all hardware building blocks for protection features;

q) setting up hardware descriptors for the virtual memory mechanism such that the privileged and unprivileged code reside in well-defined, limited, and non-overlapping memory areas;

r) refusing to perform any service on behalf of untrusted code, or to allow the attempted use to proceed, when the hardware generates a fault due to the untrusted code attempting to access a memory location outside its allowed memory area.

The enforcement of separation between the security domains of subjects in the TSC (FPT_SEP.1.2) is done partly by the hardware and partly by software. The hardware provides primitive building blocks to support this as follows:

- Has a virtual memory mechanism that allows the set of physical “pages” accessible to the current execution thread to be limited to a subset of all physical pages available to the current domain.
- The virtual-to-physical mapping allows certain areas to be marked read-only.
- The virtual-to-physical mapping is switchable only by hardware-privileged software and unprivileged software has no way to disable, circumvent, or modify the mapping.

However, the hardware does not “know”:

a) which objects and/or code constitutes a subject;

b) which constituent parts of a subject can be shared with other subjects.

Also, some aspects of providing the domain separation are implemented completely in software, with no reliance on the hardware building blocks. The TSF (software) implements the protection, with full awareness of the security state and requirements, by:

a) packaging code into file system objects (well-defined, well-bounded containers);

b) implementing process-local memory areas (in addition to process-global memory areas) that are of limited size and that do not overlap with any other memory areas;

c) placing the code and private data for a subject in process-local memory areas;

d) not providing any service to untrusted code that allows one subject to access another subject's process-local memory areas;

e) requiring TSF involvement in creation of new subjects and switching of control between subjects;

f) using the mandatory integrity mechanism to prevent modification of TSF objects (which may be "mapped" into a subject) by untrusted code;

g) setting up hardware tables in a particular way to define the memory areas for a process;

h) associating specific hardware tables with specific processes;

i) setting up hardware memory descriptors in a particular way for a subject such that the hardware will cause a fault if the subject attempts to access memory outside the bounds of the descriptors;

j) setting up all hardware fault descriptors so that the TSF will catch all faults from the hardware;

k) refusing to perform any service on behalf of untrusted code, or to allow the attempted use to proceed, when the hardware generates a fault due to a subject attempting to access memory outside the bounds allowed by the subject's hardware memory descriptors;

l) setting the CPU state to enable hardware building blocks for virtual memory support.

6.1.9.2 How Reference Validation is Provided by the TOE

The insurance that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed (FPT_RVM.1.1) is done partly by the hardware and partly by the software. The hardware provides primitive building blocks to support this are the same as those given above for domain separation.

However, the hardware does not "know":

a. which kinds of objects and functions exist;

b. which different kinds of accesses are supported.

Also, some aspects of providing the reference validation mechanism are implemented completely in software, with no reliance on the hardware building blocks. The TSF (software) implements the protection, with full awareness of the security state and requirements, by:

a) defining a set of objects, their attributes, and how the policies apply to them;

b) implementing (fully in software) access control (at every access attempt) for semaphore, device, socket, and process objects;

c) implementing (fully in software) access control during initial opening/mapping for file system and shared memory objects;

d) providing services to non-TSF software that allow well-defined portions of file system and shared memory objects to be "mapped" into well-defined areas of a process's address space;

e) setting up hardware memory descriptors to limit access to mapped file system and shared memory objects to exactly the approved regions of the process's address space;

f) requiring TSF involvement in creation of, and initial access to, all kinds of objects;

g) converting requests for "execute" permission to read permission and converting requests for "write-only" permission to read-write permission, because of limitations in the underlying abstract machine;

h) disabling hardware memory descriptors when access to mapped objects is revoked;

i) setting up all hardware fault descriptors so that the TSF will catch all faults from the hardware;

j) refusing to perform any service on behalf of untrusted code, or to allow the attempted use to proceed, when the hardware generates a fault due to a subject attempting to access an object through a disabled hardware memory descriptor;

k) setting the CPU state to enable hardware building blocks for virtual memory support.

6.1.9.3 How Reliable Time Stamps are Provided by the TOE

The production of reliable time stamps (FPT_STM.1.1) is done partly by the hardware and partly by the software. The hardware provides primitive building blocks to support this as follows:

- It stores, and updates, the time-of-day while the TOE is not running.
- It provides periodic timers/counters that the TSF can use at run-time to calculate the current time-of-day.

However, the hardware does not "know":

a) whether the stored time is correct;

b) which time zone or daylight savings time offsets to apply;

c) whether relative or absolute times are needed;

d) what format times will be kept or displayed in.

Also, some aspects of providing reliable time stamps are implemented completely in software, with no reliance on the hardware building blocks. The TSF (software) implements the protection, with full awareness of the security state and requirements, by:

a) applying appropriate time zone and daylight savings modifications to the time;

b) setting up the PIT registers to count and interrupt in a particular way;

c) setting up hardware descriptors to route PIT interrupts to the appropriate TSF interrupt handler;

d) using PIT interrupts to update the running (absolute) time stored in the TSF (software);

e) notifying the interrupt hardware when interrupt processing is complete;

f) serialising access to the PIT hardware counter and to the TSF-stored time;

g) placing time stamps in audit records, console log entries, and databases of login activity;

h) displaying time stamps associated with audit records and previous login activity.

i) providing an interface for an operator to display and change the absolute time kept in software and to change the time zone and daylight savings values;

j) placing corrected times into the hardware for storage when the TOE is not active;

k) providing a programmatic interface for non-TSF software to obtain the current absolute time.

6.1.10 Trusted Path – TRUPTH

The trusted path function, TRUPTH, provides a trusted communication path between users and the TSF.

Note that “remote” users, i.e., across a network, are not supported. Users on serial terminals are considered local users. The <Break> key invokes the Trusted Path key for serial terminal users. On the console the sequence is <Ctrl-Alt-SysRq>. These are known as the SAK (Secure Attention Key). Any invocation of the SAK leads to a Trusted Path.

SAK must be used to initiate a login. Any time SAK is used, the user will obtain a prompt from a part of the TSF known as the Secure Server. If the terminal is not already handling a login session, a login is initiated; otherwise the user can request running of any trusted command. Use of SAK while processes are already running returns the display to a known state and severs access by those processes to the display. Access to the display by those processes can be restored with the trusted “reattach” command.

6.2 Assurance Measures

The XTS-400, Version 6.4(UKE) claims to satisfy the assurance requirements for Evaluation Assurance Level (EAL) 5+.*Table 6: Security Assurance Requirements and the Security Measures That Meet Them* shows the assurance requirements and the assurance measures that meet them.

Table 6: Security Assurance Requirements and the Security Measures That Meet Them

Class Name	**Assurance Component**	**Assurance Measure(s) and Rationale**	**Document(s)**
Configuration Management	ACM_AUT.1	CVS is used to control changes to the implementation representation. Scripts and CVS provide automated support for generating the TOE.	CM-Plan
	ACM_CAP.4	Each version of the TOE has a unique version ID and that ID is available through a trusted command and appears on the delivery documentation. All constituent components of the TOE are associated with configuration items and each version of each item is uniquely identified. Processes are in place such that all changes to configuration items have to be reviewed and approved. CM processes include procedures to build a release of the TOE.	CM-Plan
	ACM_SCP.3	CM processes cover not only the implementation representation, but also documents, product flaws, and development tools. The CM system includes mechanisms to retrieve older versions of configuration items and to determine differences between versions of any configuration item.	CM-Plan
Delivery and Operation	ADO_DEL.2	Processes are in place to assure secure delivery of the TOE to customers.	CM-Plan, Software Release Bulletin
	ADO_IGS.1	Specific procedures exist for the customer to securely install and start up the TOE.	Software Release Bulletin, TFM
Development	ADV_FSP.3	The external TSF interfaces are documented.	Functional Specification
	ADV_HLD.3	The high-level design of the TSF is documented.	HLD
	ADV_IMP.2	All source code is provided.	
	ADV_INT.1	The TSF is internally designed and implemented using strong architectural features.	HLD (particularly the section of the top-level document entitled *General Architectural Features*), LLD
	ADV_LLD.1	The low-level design of the TSF is documented.	LLD
	ADV_RCR.2	Correspondence between all design abstraction levels is documented.	LLD, HLD, Functional Specification
	ADV_SPM.3	MAC, MIC, and DAC policy enforcement follows strict rules. Correspondence to the Functional Specification is documented.	Security Model

Class Name	Assurance Component	Assurance Measure(s) and Rationale	Document(s)
Guidance Documents	AGD_ADM.1	Extensive administrative guidance is provided to the customer.	TFM
	AGD_USR.1	Extensive user guidance is provided to the customer.	User's Manual
Life Cycle Support	ALC_DVS.1	Physical, procedural, and personnel processes are in place to protect the TOE while in development.	CM-Plan
	ALC_FLR.3	There are specific points-of-contact for a customer to use when reporting TOE problems. There is a bug reporting and tracking tool and a database of bug reports. Processes are in place to quickly recognize security flaws and to remedy them.	CM-Plan
	ALC_LCD.2	A standard life-cycle model governs changes made to the TOE.	CM-Plan
	ALC_TAT.2	The CM system defines the development tools. The Coding Standards provide a reference for meaning of implementation representation statements and shows the values of tools options.	CM-Plan, Coding Standards, Documentation Guidelines
Test	ATE_COV.2	Multiple tests are provided for each TSF interface.	Test Guide
	ATE_DPT.2	Many tests indirectly test each subsystem. A depth analysis shows which tests map to each subsystem and module.	Test Guide
	ATE_FUN.1	The purpose, method, and expected results of each test are documented.	Test Guide
	ATE_IND.3	Method for installing and running the tests is documented.	Test Guide
Vulnerability Analysis	AVA_CCA.1	Covert channel analysis has been performed and the capacities of all usable channels are estimated.	Covert Channel Analysis
	Assurance Component	**Assurance Measure(s) and Rationale**	**Document(s)**
	AVA_MSU.2	The TFM documents all modes of operation, all assumptions about the IT environment, and mentions all requirements for external security measures. An analysis has been performed to verify that the guidance is complete, consistent, and clear.	TFM
	AVA_SOF.1	A strength-of-function analysis has been performed on password guessing. The strengths of these mechanisms meet the requirements of this ST and the LSPP.	ST
	AVA_VLA.3	A systematic search for vulnerabilities has been made.	Vulnerability Analysis

6.3 IT Security Functions Realised by Probabilistic or Permutational Mechanisms

The only IT security function that includes probabilistic or permutational mechanisms is IDNAUT. As stated in section *5.2*, this TOE meets a strength of SOF-high.

Within IDNAUT, the methods used to provide difficult-to-guess passwords are probabilistic.

This page is intentionally left blank

Section 7 PP Claims

This section provides the PP Conformance claims for the XTS-400, Version 6.4(UKE) Security Target.

7.1 PP Reference

The TOE conforms to the Labeled Security Protection Profile (LSPP) Version1.b, 8 October, 1999. It also conforms to the Controlled Access Protection Profile (CAPP) Version 1.d, 8 October, 1999. CAPP is a subset of LSPP and will not be discussed further.

7.2 PP Refinements, Selections, and Assignments

Refinements, selections, and assignments to the requirements from the LSPP are shown in boldface in Section *5.1* of this ST. *Table 2: Security Functional Requirements* denotes which SFRs are part of the LSPP.

7.3 PP Additions

Assumptions, Threats, and Objectives have been added or further refined to reflect the capabilities of the TOE.

The following objectives were added:

- O.AC_Admin_Limit
- O.Access_History
- O.Admin_Role
- O.Audit_Protection
- O.Authorities
- O.Display_Banners
- O.Mandatory_Integrity
- O.Markings
- O.Protect
- O.Recovery
- O.Self_Protection
- O.Testing

- O.Trained_Users
- O.Trusted_Path
- O.User_Authentication
- O.User_Identification
- O.Vulnerability_Analysis.

The following SFRs have been added to support these objectives:

- FAU_SAA.1
- FAU_STG.2 (LSPP requires only FAU_STG.1)
- FDP_ACC.2 (LSPP requires only FDP_ACC.1)
- FDP_IFC.2 (LSPP requires only FDP_IFC.1)
- FIA_AFL.1
- FIA_UAU.2 (LSPP requires only FIA_UAU.1)
- FIA_UAU.4
- FIA_UID.2 (LSPP requires only FIA_UID.1)
- FMT_MOF.1
- FMT_MSA.2
- FMT_MTD.2
- FMT_SAE.1
- FMT_SMF.1
- FMT_SMR.3
- FPT_ RCV.1
- FPT_TST.1
- FTA_SSL.1
- FTA_SSL.2
- FTA_SSL.3
- FTA_TAB.1
- FTA_TAH.1
- FTA_TSE.1

- FTP_TRP.1

The LSPP uses an EAL3 assurance package. Therefore all of those assurance requirements are included or superseded by this ST (EAL5+). EAL5 includes the following assurance components that are not even part of EAL3: ACM_AUT, ADV_IMP, ADV_INT, ADV_LLD, ADV_SPM, ALC_LCD, ALC_TAT, and AVA_CCA. In addition, this ST includes ALC_FLR, which is not even part of EAL5.

The additional SFRs provide additional features desired by XTS-400 customers, make the system more flexible for a wider range of customers and applications, and contribute to overall higher security. The additional assurance is needed by customers that have systems in higher threat environments and with more levels of more sensitive data than is intended by the LSPP.

This page is intentionally left blank

Section 8 Rationale

This section of the ST will describe how the threats, assumptions, and policies are addressed by the security objectives of the TOE. *Table 7: Mapping the TOE Security Environment to Security Objectives* provides the mapping of each threat, assumption, and policy to the security objective(s) that address them.

For a rationale that each security function defined in the TOE Summary Specification is suitable to meet the SFRs, see *Table 5: SFRs and the Security Functions that Satisfy Them*. For a rationale that each assurance measure defined in the TOE Summary Specification is suitable to meet the SARs, see *Table 6: Security Assurance Requirements and the Security Measures That Meet Them*.

8.1 Security Objectives Rationale

Table 7: Mapping the TOE Security Environment to Security Objectives

Policy/Threat/Assumption	Justification	Objective
	Assumptions	
A.Admin_Errors	OE.Admin ensures that only competent administrators are chosen and they understand the guidance documentation.	OE.Admin
A.Outsider_Hi	Many of the objectives for both the TOE environment are intended to counter a highly capable and motivated attacker. Specifically, the environment must physically protect the system and must provide for capable administrators that will correctly configure the system and review system activity. Other mechanisms are provided by the TOE itself.	OE.Admin OE.Physical
A.User_Mistakes	OE.User ensures that only competent users are chosen and they understand the guidance documentation.	OE.User

Policy/Threat/Assumption	Justification	Objective
	Policies	
P.Accountability	Auditing security relevant events provides a basis for user accountability. Administrators must exist to enable and review the auditing.	O.Admin_Role O.Audit_Gen_User, O.Audit_Generation, O.Audit_Protection,
P.Authorities	Audit data would provide a basis for threats being reported to authorities. The vendor also has a flaw remediation process in place.	O.Audit_Gen_User, O.Audit_Review, O.Admin_Trained, O.Authorities
P.Authorized_Use	Information shall be used for authorized purposes through training, System Banners and through audit accountability. Administrators must exist to configure the mandatory levels of user terminals and management functions must exist to allow such configuration.	O.Admin_Role O.Display_Banners O.Audit_Gen_User O.Audit_Review O.Manage O.Trained_Users O.Admin_Trained
P.Authorized_Users	The TOE shall ensure that only authorized users gain access to the TOE and its users. Users are also told at login time of previous logins so that they may assess if another user has been using their account. Administrators must exist to add and remove users from the system and management functions must exist to allow such configuration.	O.Access O.Access_History O.Admin_Role O.Manage
P.Availability	The TOE shall protect resources to ensure information availability to authorized users and processes.	O.Self_Protection O.Trusted_System_Operation
P.Classification	The TOE shall restrict access by sensitivity level. Administrators must exist to configure the MAC levels of users and system devices and management functions must exist to allow such configuration.	O.Admin_Role O.Mandatory_Access O.Manage
P.Guidance	The TOE shall provide written procedures to ensure proper installation and operation by trained users and administrators.	O.Install O.Admin_Trained O.Trained_Users

Policy/Threat/Assumption	Justification	Objective
P.Information_AC	Information access shall be limited to authorized users and processes by the MIC, MAC, DAC policy enforcement and by audit traceability	O.Audit_Gen_User O.Discretionary_Access O.Discretionary_User_Control O.Mandatory_Access O.Mandatory_Integrity
P.Integrity	Information integrity shall be protected by the TSF's enforcement of MIC policy.	O.Mandatory_Integrity O.Trusted_System_Operation
P.Marking	Data contained in named objects shall have associated MAC, MIC and DAC labels. Unnamed objects shall protect information by inheriting the owning subject's sensitivity/integrity levels.	O.Discretionary_Access O.Discretionary_User_Control O.Mandatory_Access O.Mandatory_Integrity O.Markings
P.Need_to_Know	Users must have a need to know before accessing or destroying data. Administrators must exist to configure the group membership and cleared security categories for users and management functions must exist to allow such configuration.	O.Admin_Role O.Discretionary_Access O.Manage O.Mandatory_Access O.Mandatory_Integrity
	Threats	
T.Malicious_Code	Malicious code is prevented from executing on the TOE by resource protection, and MAC, MIC, DAC policy enforcement.	O.Self_Protection O.Protect O.Discretionary_Access O.Discretionary_User_Control O.Mandatory_Access O.Mandatory_Integrity
T.Admin_Err_Commit	To prevent errors of administrators that compromise security, they are trained properly and limited in roles and access.	O.Admin_Trained O.AC_Admin_Limit
T.Admin_Err_Omit	To prevent errors of administrators that compromise security, they are trained properly.	O.Admin_Trained
T.Admin_Hostile_Modify	The threat of a hostile administrator is mitigated through role and access limitation and traceability by audit.	O.AC_Admin_Limit, O.Audit_Gen_User

Policy/Threat/Assumption	Justification	Objective
T.Covert_Channel	The threat of covert channel use is mitigated by design of the TSF to eliminate some of them, auditing of them where possible, and timing delays to slow their speed. The MAC policy mechanism involves covert channel reduction mechanisms. These mechanisms and the scenarios of actual channels are documented via AVA_CCA.1.	O.Audit_Generation O.Covert_Channel_Reduction O.Mandatory_Access O.Trusted_System_Operation
T.Dev_Flawed_Code	The threat of flawed code is mitigated by configuration management, rigorous testing and vulnerability analysis.	O.Config_Mgmt O.Testing O.Vulnerability_Analysis
T.Hack_AC	The threat of a hacker attack is mitigated by strict enforcement of MAC, MIC, DAC policies, resource protection, as well as by ensuring the system can withstand penetration and other attempts to gain unauthorized access. Users are also told at login time of previous logins so that they may assess if another user has been using their account.	O.Audit_Gen_User O.Access O.Access_History O.Audit_Generation O.Audit_Protection O.Covert_Channel_Reduction O.Discretionary_Access O.Discretionary_User_Control O.Mandatory_Access O.Mandatory_Integrity O.Protect O.Self_Protection O.Trusted_Path O.Vulnerability_Analysis
T.Audit_Corrupt	Audit files are to be well protected and accessible to authorized administrators.	O.Audit_Protection O.Mandatory_Access O.Mandatory_Integrity O.User_Identification O.User_Authentication
T.Config_Corrupt	Configuration files are to be well protected and accessible to authorized administrators.	O.Mandatory_Access O.Mandatory_Integrity O.User_Identification O.User_Authentication
T.Improper_Installation	The TOE shall be installed in accordance with well defined procedures by trained individuals.	O.Install O.Admin_Trained

Policy/Threat/Assumption	Justification	Objective
T.Insecure_Start	The TOE shall be able to restart without compromising security.	O.Recovery
T.Objects_Not_Clean	The TSF will ensure that information stored in an object will not be shared if object resource is reallocated.	O.Residual_Information O.Trusted_System_Operation
T.Poor_Design	This threat will be countered by sound, proven, accepted design practices.	O.Sound_Design O.Vulnerability_Analysis
T.Poor_Implementation	This threat will be countered by a sound implementation of the design.	O.Sound_Implementation O.Testing O.Vulnerability_Analysis
T.Poor_Test	The TOE will undergo rigorous testing to ensure that all security objectives are met.	O.Testing
T.Power_Disrupt	This threat is mitigated by trusted recovery.	O.Recovery
T.Replay	This threat is countered by performing authentication over a trusted path.	O.Trusted_Path O.Trusted_System_Operation
T.Sysacc	The threat of a malicious user or process gaining administrator level access is mitigated by strict enforcement of MAC, MIC, DAC policies, resource protection, as well as by ensuring the system can withstand penetration and other attempts to gain unauthorized access.	O.Audit_Gen_User O.Access O.Audit_Generation O.Audit_Protection O.Covert_Channel_Reduction O.Discretionary_Access O.Discretionary_User_Control O.Mandatory_Access O.Mandatory_Integrity O.Protect O.Self_Protection O.Trusted_Path O.Vulnerability_Analysis
T.Unattended_Session	The TOE implement measures to ensure unattended sessions cannot be exploited by malicious users or processes	O.Access OE.Secure_Term

Policy/Threat/Assumption	Justification	Objective
T.Unauth_Modification	The threat of unauthorized modifications to system attributes and resources is mitigated through the enforcement of MAC, MIC, DAC policies, resource protection, as well as by ensuring the system can withstand penetration and other attempts to gain unauthorized access.	O.Audit_Gen_User O.Access O.Audit_Generation O.Audit_Protection O.Covert_Channel_Reduction O.Discretionary_Access O.Discretionary_User_Control O.Mandatory_Access O.Mandatory_Integrity O.Protect O.Self_Protection O.Vulnerability_Analysis
T.User_Corrupt	The threat of unauthorized modifications to system attributes and resources is mitigated through the enforcement of MAC, MIC, DAC policies, resource protection, as well as by ensuring the system can withstand penetration and other attempts to gain unauthorized access.	O.Audit_Gen_User O.Access O.Audit_Generation O.Audit_Protection O.Discretionary_Access O.Discretionary_User_Control O.Mandatory_Access O.Mandatory_Integrity O.Protect O.Self_Protection O.Vulnerability_Analysis
T.Hack_Social_Engineer	This threat is mitigated by the proper training of administrators and users regarding password selection. Users are also told at login time of previous logins so that they may assess if another user has been using their account.	O.Trained_Users O.Admin_Trained O.Access O.Access_History O.User_Authentication
T.Spoofing	The threat of spoofing from another machine is mitigated by the use of the Trusted Path. Users are also told at login time of previous logins so that they may assess if another user has been using their account.	O.Access_History O.Trusted_Path

Policy/Threat/Assumption	Justification	Objective
T.User_Improper_Export	This threat is mitigated through the use of MAC, MIC, DAC policy enforcement and audit traceability.	O.Audit_Gen_User O.Discretionary_Access O.Discretionary_User_Control O.Mandatory_Access O.Mandatory_Integrity
T.User_Abuse	This threat is mitigated through audit traceability.	O.Audit_Gen_User
T.User_Error	This threat is mitigated through the enforcement of MAC, MIC, DAC policy; user and administrator training; and policy and usage warnings and reminders on the system access banner.	O.Audit_Gen_User O.Trained_Users O.Admin_Trained O.Discretionary_Access O.Discretionary_User_Control O.Display_Banners O.Mandatory_Access O.Mandatory_Integrity
T.Trusted_User_Error	The TOE provides self-protection by maintaining a domain for its own execution that protects itself and its resources from normal user interference and tampering. Specifically, the 4 ring software architecture described in detail in section 2.1.1.1 isolates users and processes with low privilege from trusted system functions.	O.Self_Protection

Table 8: Tracing of Security Objectives to the TOE Security Environment

Assumptions/Policies/Threats	Justification	Objectives
A.Acc_to_Comms	Physical security of communication to TOE.	OE.Physical
A.Admin_Docs	The TOE itself cannot govern behaviour of administrators.	OE.Admin
A.Clearance	The environment must provide procedures for determining how much “trust” (in terms of MAC and MIC clearance, group membership, XTS capabilities, and access to console), and which needs-to-know, to place on each system user.	OE.Clearance

Assumptions/Policies/Threats	Justification	Objectives
A.Competent_Admin	The TOE itself can not determine whether administrators are competent. This objective includes selection of competent administrators.	OE.Admin
A.Coop_User	The TOE itself can not determine whether users will be cooperative. This objective includes procedures for selecting, training, and monitoring cooperative users.	OE.User
A.Dispose_User_Data	The TOE itself can not ensure that administrators follow procedures. This objective includes selection, training, and monitoring of competent administrators and one of the functions of an administrator is to properly dispose of user data.	OE.Admin
A.Handling_of_Data	The site is assumed to have procedures for how sensitive, classified, and high-integrity data and secrets are to be handled when they are in possession of an authorized user. The site is also assumed to have procedures for pickup and distribution of hardcopy output at multi-user or multi-level printers.	OE.Handling_of_Data
A.No_Trusted_Network	The TOE itself can not determine what kinds of information other systems on the network transmit and, therefore, the physical connection must gain assurance of its MAC and MIC level by being under the same management regime as the TOE.	OE.Connection_Mgmt
A.Password_Management	The TOE itself can not ensure that administrators follow procedures. This objective includes selection, training, and monitoring of competent administrators. Training of administrators includes ensuring that they read and understand the TFM, which includes instructions on password management.	OE.Admin
A.Phys_Acs_to_Out	The TOE is assumed to have adequate physical security protection	OE.Physical

Assumptions/Policies/Threats	Justification	Objectives
A.Physical	The TOE is assumed to have adequate physical security protection	OE.Physical
A.Protect_From_Out	The TOE is assumed to have adequate physical security protection	OE.Physical
A.Review_Audit_Log	The TOE itself can not ensure that administrators follow procedures. This objective includes selection, training, and monitoring of competent administrators. Training of administrators includes ensuring that they read and understand the TFM, which includes instructions on using the auditing mechanism.	OE.Admin
A.Secure_Term	The TOE itself can not govern user behavior and can not control the local storage of any arbitrary terminal. This objective includes procedures for selecting terminals that meet the assumptions of the TOE, documenting for users how to securely use the terminals, and training the users.	OE.Secure_Term
A.Sensitivity	The TOE itself can not determine the proper security attributes of imported data. This objective includes procedures for specifying a consistent sensitivity labeling scheme, for determining the correct sensitivity of information, for determining the allowed clearance of users, and for handling of sensitive material.	OE.Sensitivity
P.Physical_Control	The TOE is assumed to have adequate physical security protection	OE.Physical

Assumptions/Policies/Threats	Justification	Objectives
T.Hack_Phys	This threat is mitigated by physical security.	OE.Physical

8.2 Security Requirements Rationale

8.2.1 Functional Security Requirements Rationale

Table 9: Functional and Assurance Component to Security Objective Mapping

Objectives	Justification	Requirements
O.AC_Admin_Limit	The administrative roles are limited by identification and authentication, MAC, MIC and DAC privileges, and audit traceability.	FAU_GEN.1, FAU_GEN.2, FAU_SAR.1, FAU_SAR.2, FAU_SAR.3, FAU_SEL.1, FAU_STG.2, FAU_STG.4, FDP_IFC.2, FDP_IFF.2, FDP_RIP.2, FIA_AFL.1-N, FIA_ATD.1, FIA_SOS.1, FIA_UAU.2, FIA_UAU.4, FIA_UAU.7
O.Audit_Gen_User	A requirement for a comprehensive audit capability meets this objective.	FAU_GEN.1, FAU_GEN.2, FAU_SAA.1
O.Access	Authorized access is ensured through the use of identification and authentication of users, and control of user privileges. Passwords can expire to limit damage done with stolen or cracked passwords and individual terminals can restrict the access range within which they can be used. Reuse of an open user session by a second user while the first user is away from a terminal is mitigated by mechanisms to lock or terminate inactive sessions.	FAU_SAA.1, FIA_AFL.1, FIA_ATD.1, FIA_SOS.1, FIA_UAU.2, FIA_UAU.4, FIA_UAU.7, FIA_USB.1, FTA_SSL.1, FTA_SSL.2, FTA_SSL.3, FTA_TSE.1

Objectives	Justification	Requirements
O.Access_History	This requirement's definition specifically meet's this objective.	FTA_TAH.1
O.Admin_Role	This objective is accomplished through identification and authentication of administrator, and by limiting administrative functions to highly privileged administrator(s).	FAU_SEL.1, FAU_STG.2, FDP_IFC.2, FIA_AFL.1, FIA_ATD.1, FIA_SOS.1, FIA_UAU.2, FIA_UAU.4, FIA_UAU.7, FIA_UID.2, FMT_MOF.1, FMT_MSA.1, FMT_MTD.1, FMT_MTD.2, FMT_SMR.1, FMT_SMR.3
O.Admin_Trained	The TOE developer will provide adequate documentation to ensure secure management by the administrator(s).	AGD_ADM.1
O.Audit_Generation	A requirement for a comprehensive audit capability meets this objective. The reliable time stamp mechanism is used to place accurate time stamps on audit records.	FAU_GEN.1, FAU_GEN.2, FPT_STM.1
O.Audit_Protection	These requirements specify audit data protection.	FAU_STG.2, FAU_STG.3, FAU_STG.4
O.Audit_Review	These requirements specify that the administrator will have audit capability and tools to query audit data.	FAU_SAA.1, FAU_SAR.1, FAU_SAR.2, FAU_SAR.3
O.Authorities	The TOE developer will have a flaw remediation process in place.	ALC_FLR.3
O.Config_Mgmt	This assurance requirement specifically details configuration management requirements for the TOE.	ACM_AUT.1, ACM_CAP.4, ACM_SCP.3
O.Covert_Channel_Reduction	This requirement specifically details covert channel analysis assurance requirements of the TSF.	AVA_CCA.1

Objectives	Justification	Requirements
O.Discretionary_Access	These requirements specifically delineate TOE responsibilities regarding discretionary access controls (DAC).	FDP_ACC.2, FDP_ACF.1, FDP_ETC.1, FDP_ITC.1, FDP_ITC.2
O.Discretionary_User_Control	These requirements specifically delineate TOE responsibilities regarding discretionary access controls (DAC).	FDP_ACC.2, FDP_ACF.1
O.Display_Banners	This requirement is to provide banner displays in accordance with the objective.	FTA_TAB.1
O.Install	This assurance requirement specifically details documentation required for installation activities.	ADO_IGS.1
O.Manage	These security requirements detail all of the functions required to allow the administrator to properly manage the security of the TOE.	FMT_MOF.1, FMT_MSA.1, FMT_MSA.2, FMT_MSA.3, FMT_MTD.1, FMT_MTD.2, FMT_REV.1, FMT_SAE.1, FMT_SMF.1, FMT_SMR.1, FMT_SMR.3
O.Mandatory_Access	These requirements detail the mandatory access control requirements of the TSF.	FDP_ETC.1, FDP_IFC.2, FDP_IFF.2, FDP_ITC.1, FDP_ITC.2
O.Mandatory_Integrity	These requirements detail the mandatory integrity control requirements of the TSF.	FDP_IFC.2, FDP_IFF.2
O.Markings	This requirement details the printer output labeling requirements of the TOE.	FDP_ETC.2
O.Protect	Data and resources are protected by the enforcement of MAC, MIC, DAC policies.	FDP_ACC.2, FDP_ACF.1, FDP_IFC.2, FDP_IFF.2
O.Recovery	The TOE shall have the functionality to allow for recovery after a system malfunction.	FPT_RCV.1
O.Residual_Information	The TOE shall prevent users or processes from accessing residual information upon reallocation of resources.	FDP_RIP.2

Objectives	Justification	Requirements
O.Self_Protection	These requirements relate to maintenance of separate domains for the TSF and non-TSF entities and for testing to ensure that the TSF mechanisms are still working.	FPT_AMT.1, FPT_SEP.1, FPT_TST.1
O.Sound_Design	This assurance requirement details the requirements of the TOE design to ensure that the security objectives are met.	ADV_HLD.3
O.Sound_Implementation	This assurance requirement details the requirement of the TOE Implementation to ensure that the security objectives are met.	ADV_IMP.2, ALC_FLR.3
O.Testing	This assurance requirements details the requirement of independent testing that ensures that the function requirements have been met.	ATE_COV.2, ATE_DPT.2, ATE_FUN.1, ATE_IND.3, AVA_VLA.3
O.Trained_Users	The TOE developer will provide adequate documentation to ensure secure operation by users.	AGD_USR.1
O.Trusted_Path	The TOE will be required to provide a trusted path to remote and local users.	FTP_TRP.1
O.Trusted_System_Operation	The TOE must ensure TSF is not bypassed.	FPT_RVM.1, FPT_SEP.1 FPT_STM.1
O.User_Authentication	The TOE shall require users to be authenticated before any access to TSF mediated actions.	FIA_UAU.2, FIA_UAU.4, FIA_UAU.7, FAU_SAA.1
O.User_Identification	The TOE shall require users to be identified before any access to TSF mediated actions.	FIA_UID.2
O.Vulnerability_Analysis	The TOE shall undergo a comprehensive vulnerability analysis as described in these assurance requirements.	AVA_CCA.1, AVA_MSU.2, AVA_SOF.1, AVA_VLA.3
OE.Physical		environment
OE.Clearance		environment
OE.Handling_of_Data		environment

Objectives	Justification	Requirements
OE.Sensitivity		environment

8.2.2 Assurance Security Requirements Rationale

EAL3 is required by the LSPP. EAL5+ level of assurance was chosen for this ST because the product is intended to be used in hostile environment while protecting very sensitive information. EAL4 products are not required to be designed with anywhere close to the same level of assurance of that is designed into the XTS-400. Finally, EAL5 meets or exceeds all “medium robustness” requirements, except for covert channel analysis of cryptographic modules.

The XTS-400 is used in very sensitive environments and its customers want every ounce of assurance they can get.

The following assurance requirements have augmented the basic EAL5 package:

ALC_FLR.3

ATE_IND.3

Addition of ALC_FLR.3 was natural because BAE -IT already had flaw remediation mechanisms in place. Some newer PPs are also requiring ALC_FLR. BAE -IT intends for the XTS-400 product to enter an assurance maintenance phase in the future.

Addition of ATE_IND.3 was natural because BAE -IT already has a security test suite which is designed to be run in entirety by evaluators.

Though the TOE could easily meet certain EAL6 requirements, significant additional staff-hours (and lab costs) would be needed to meet some of the EAL6 documentation and testing requirements. Time to market of the evaluated TOE would also be delayed if an EAL6 evaluation were performed.

8.2.3 Rationale that IT Security Requirements are Internally Consistent

There are no requirements that conflict with one another. The MAC, MIC, and DAC policies (see FDP_ACC.2, FDP_ACF.1, FDP_IFC.2, and FDP_IFF.2) operate on an overlapping set of objects and operations, but their attributes are overlapping and additive, and the rules and operations are orthogonal. The processing order is MAC, MIC then DAC – the order cannot be changed. ADV_SPM provides formal modeling of these policies and AGD_USR provides user-level documentation of these policies. Import and export of data, with or without security attributes, (see FDP_ETC.1, FDP_ITC.1, FDP_ETC.2 and FDP_ITC.2) is just a special case of these policies as applied to device objects.

All objects are also subject to the residual data mechanisms (see FDP_RIP.2). The residual data mechanisms are orthogonal to the policy mechanisms.

The MAC policy mechanism involves covert channel reduction mechanisms. These mechanisms and the scenarios of actual channels are documented via AVA_CCA.1.

The MIC policy mechanism underpins the role mechanism (see FMT_SMR.1). The role mechanism does not modify the MIC policy attributes or rules.

Login processing brings in elements of many requirements, but all in a complementary way. The login starts with an invocation of the trusted path mechanism (see FTP_TRP.1). FIA_UID.2 wants the user identified before allowing any other operations and FIA_UAU.2 wants the user authenticated before allowing any other operations. FIA_SOS defines the strength of the authentication. FTA_TSE allows for a login to be denied based on several conditions. FIA_UAU.7 requires that feedback from authentication input be obscured and FTA_TAH requires a history of previous logins, but this history does not have to include authentication values. FAU_SAA and FIA_AFL require that the number of failed login attempts be recorded and limited. Initial MAC,

MIC, and DAC attributes are imparted on the user's session according to FIA_USB. FTA_TAB requires that a banner be displayable upon successful login.

Audit records generally contain a lot of detail (see FAU_GEN.1.2), but in specific cases, that detail is not meaningful. The subject identity is not meaningful for an authentication failure nor for records generated internally by the TSF (such as to show the startup of the audit functions). FAU_SEL.1.1 generally requires that the audit trail be searchable by subject identity, the naturally the aforementioned records would not reliably show up in such a search. Passwords are not kept in login audit records, but there is no requirement that they should be.

Audit records are generated for many events where other requirements are coming to bear, such as login, policy check failures, and system recovery. The audit record generation mechanism is orthogonal to these other mechanisms.

The management requirements (FMT_) are related to many of the mechanisms involved with other requirements. In many cases, the other mechanisms will enforce the settings made through management functions. Installation mechanisms (see ADO_IGS.1) rely on management functions. The administrator guidance (see AGD_ADM) documents the management functions.

Use of many of the management functions relies on a use of the trusted path mechanism.

8.3 Functional Requirements Grounding in Objectives

Table 10: Requirements to Objectives Mapping

Requirements	Justification	Objectives
FAU_GEN.1	The TSF shall be able to generate audit events to satisfy these objectives.	O.AC_Admin_Limit, O.Audit_Gen_User, O.Audit_Generation
FAU_GEN.2	The TSF shall provide detailed audit information to accomplish these objectives.	O.AC_Admin_Limit, O.Audit_Gen_User, O.Audit_Generation
FAU_SAA.1	The TSF shall detect an accumulation of failed login attempts to help satisfy these objectives.	O.Access, O.Audit_Gen_User, O.Audit_Review, O.User_Authentication
FAU_SAR.1	The TSF shall provide specific high level audit information to satisfy these objectives.	O.AC_Admin_Limit, O.Audit_Review

Requirements	Justification	Objectives
FAU_SAR.2	Audit review shall be restricted to designated administrators.	O.AC_Admin_Limit, O.Audit_Review
FAU_SAR.3	Audit record querying functions shall be made available to the administrator.	O.AC_Admin_Limit, O.Audit_Review
FAU_SEL.1	The audit administrator shall be able to select audit events.	O.AC_Admin_Limit, O.Admin_Role
FAU_STG.2	The audit data shall be protected and not accessible by any user except the designated administrator.	O.AC_Admin_Limit, O.Admin_Role, O.Audit_Protection
FAU_STG.3	The audit trail shall be protected from disk space runout or hard crashes.	O.Audit_Protection
FAU_STG.4	The audit trail shall be protected by shutting down the system when a predetermined threshold has been met.	O.AC_Admin_Limit, O.Audit_Protection
FDP_ACC.2	Discretionary Access Control Policy shall be enforced by the TSF.	O.Discretionary_Access, O.Discretionary_User_Control, O.Protect
FDP_ACF.1	Discretionary Access Control policy shall be enforced by the TSF.	O.Discretionary_Access, O.Discretionary_User_Control, O.Protect
FDP_ETC.1	Discretionary and Mandatory access control shall encompass export of data.	O.Discretionary_Access, O.Mandatory_Access
FDP_ETC.2	Printer output shall be labeled with sensitivity information.	O.Markings
FDP_IFC.2	The TSF shall enforce MAC, MIC policy.	O.AC_Admin_Limit, O.Admin_Role, O.Mandatory_Access, O.Mandatory_Integrity, O.Protect
FDP_IFF.2	The MAC and MIC policies have a well-defined set of attributes and rules. Even administrators must work within these rules, except where specific trusted commands allow specific kinds of exemptions.	O.AC_Admin_Limit, O.Mandatory_Access, O.Mandatory_Integrity, O.Protect

Requirements	Justification	Objectives
FDP_ITC.1	Discretionary and Mandatory access control shall encompass import of data.	O.Discretionary_Access, O.Mandatory_Access
FDP_ITC.2	Discretionary and Mandatory access control shall encompass import of data.	O.Discretionary_Access, O.Mandatory_Access
FDP_RIP.2	The TSF shall prevent access of residual information after reallocation of resources.	O.AC_Admin_Limit, O.Residual_Information
FIA_AFL.1	Only authorized users and administrators will have access to the TSF.	O.AC_Admin_Limit, O.Access, O.Admin_Role
FIA_ATD.1	The TSF shall maintain the list of user security attributes.	O.AC_Admin_Limit, O.Access, O.Admin_Role
FIA_SOS.1	The TSF shall enforce minimum password lengths.	O.AC_Admin_Limit, O.Access, O.Admin_Role
FIA_UAU.2	The TSF shall mediate no actions prior to user authentication.	O.AC_Admin_Limit, O.Access, O.Admin_Role, O.User_Authentication
FIA_UAU.4	This requirement further reinforces FIA_UAU.2 to meet these objectives.	O.AC_Admin_Limit, O.Access, O.Admin_Role, O.User_Authentication
FIA_UAU.7	This requirement further reinforces FIA_UAU.2 to meet these objectives.	O.AC_Admin_Limit, O.Access, O.Admin_Role, O.User_Authentication
FIA_UID.2	The TSF shall take no action before the user is identified.	O.Admin_Role, O.User_Identification
FIA_USB.1	The TSF shall allow a subject no more privilege than the user on behalf of whom the subject is executing.	O.Access
FMT_MOF.1	The TSF shall provide functionality to ensure the administrator can manage the TOE security effectively.	O.Admin_Role, O.Manage
FMT_MSA.1	The TSF shall provide functionality to ensure the administrator can manage the TOE security effectively.	O.Admin_Role, O.Manage

Requirements	Justification	Objectives
FMT_MSA.2	The TSF shall provide functionality to ensure the administrator can manage the TOE security effectively.	O.Manage
FMT_MSA.3	The TSF shall provide functionality to ensure the administrator can manage the TOE security effectively.	O.Manage
FMT_MTD.1	The TSF shall provide functionality to ensure the administrator can manage the TOE security effectively.	O.Admin_Role, O.Manage
FMT_MTD.2	The TSF shall provide functionality to ensure the administrator can manage the TOE security effectively.	O.Admin_Role, O.Manage
FMT_REV.1	The TSF shall provide functionality to ensure the administrator can manage the TOE security effectively.	O.Manage
FMT_SAE.1	The TSF shall provide functionality to ensure the administrator can manage the TOE security effectively.	O.Manage
FMT_SMF.1	The TSF shall provide functionality to ensure the administrator can manage the TOE security effectively.	O.Manage
FMT_SMR.1	The TSF shall provide functionality to ensure the administrator can manage the TOE security effectively.	O.Manage, O.Admin_Role
FMT_SMR.3	The TSF shall provide functionality to ensure the administrator can manage the TOE security effectively.	O.Manage, O.Admin_Role

Requirements	Justification	Objectives
FPT_AMT.1	The TOE shall have functionality to be able to ensure its security features are operating correctly. Some of the security features rely on the hardware, regardless of whether the hardware is part of the TSF or part of the abstract machine.	O.Self_Protection
FPT_RCV.1	The TOE shall be able to be manually recovered upon system failure without compromising security.	O.Recovery
FPT_RVM.1	The inability to bypass the TSF is essential to trusted system operations.	O.Trusted_System_Operation
FPT_SEP.1	The ability to maintain data separation is essential to trusted system operations.	O.Trusted_System_Operation, O.Self_Protection
FPT_STM.1	The ability to provide reliable time stamps is essential to trusted system operations and the generation of audit data.	O.Trusted_System_Operation, O.Audit_Generation
FPT_TST.1	The TOE's ability to test itself ensures that it can protect its own resources.	O.Self_Protection
FTA_SSL.1	The requirement for locking an interactive session after a designated time of activity helps the TOE restrict access to system resources.	O.Access
FTA_SSL.2	The requirement for allowing an interactive session to be explicitly locked by the user helps the TOE restrict access to system resources.	O.Access
FTA_SSL.3	The requirement for terminating an interactive session after a designated time of activity helps the TOE restrict access to system resources.	O.Access

Requirements	Justification	Objectives
FTA_TAB.1	This requirement fully meets the intent of the objective to have warning and advisory messages to users.	O.Display_Banners
FTA_TAH.1	The requirement fully meets the intent of the objective to have access history displayed to users.	O.Access_History
FTA_TSE.1	The requirement for denying a session due to certain conditions helps the TOE restrict access to system resources.	O.Access
FTP_TRP.1	The TOE is required to be able to establish a trusted path between local and remote users to the TSF.	O.Trusted_Path

8.4 PP Claims Rationale

This ST, although not written directly for the LSPP nor the CAPP, is in conformance with them. The functional and assurance requirements in this ST go beyond what is required by those PPs.

There are no other official PPs which are a good fit for the TOE. In particular, though the TOE could meet assurance requirements in the “Medium Robustness” OS PP, significant additional staff-time (and lab cost) would be required to meet the cryptographic functional requirements in that PP.

Because this ST is designed to support more functional and assurance requirements than the LSPP, this ST has additional security objectives to those in the LSPP. Those additional objectives can easily be seen by viewing the Security Objectives section of this ST. A mapping of the LSPP objectives to objectives in this ST is shown below.

Table 11: LSPP Objectives to ST Objectives Mapping

LSPP Objectives	ST Objectives
O.AUTHORIZATION	O.Access
O.DISCRETIONARY_ACCESS	O.Discretionary_Access, O.Discretionary_User_Control
O.MANDATORY_ACCESS	O.Mandatory_Access
O.AUDITING	O.Audit_Gen_User, O.Audit_Generation, O.Audit_Review
O.RESIDUAL_INFORMATION	O.Residual_Information
O.MANAGE	O.Manage

LSPP Objectives	ST Objectives
O.ENFORCEMENT	O.Config_Mgmt, O.Sound_Design, O.Sound_Implementation
O.INSTALL	O.Admin_Trained, O.Install, O.Manage, O.Trained_Users, O.Trusted_System_Operation
O.PHYSICAL	OE.Physical
O.CREDEN	OE.Handling_of_Data

8.5 Strength-of-Function Rationale

As stated in section *6.3*, there are some security functions based on probabilistic methods. The strength of this TOE, SOF-high, surpasses what is required by the LSPP (SOF¬medium). A strength of SOF-high is consistent with protecting the TOE's assets from highly skilled and motivated attackers (A.Outsider_Hi). See section *5.4* for the objectives that SOF supports.

The specific "strength" required of the methods used to provide difficult-to-guess passwords is stated in FIA_SOS.1. That specific strength surpasses what is given in the LSPP for the strength of the authentication mechanism. An analysis of the strength of that mechanism, as required by AVA_SOF.1, follows.

For a worst-case calculation, we assume that the administrator has configured the minimum password length to 6 characters and has not specified that the passwords contain mixed-case, contain non-alphabetic characters, or be subjected to dictionary checks. That creates a password space of 26 to the 6th power, which equals 3.09e08. If we then assume that humans would tend to construct passwords from only 1% of the password space, we are still left with a random guess having only a 1 in 3,090,000 chance of matching a user's actual password. The LSPP, which is SOF-medium, requires only a 1 in 1e6 chance.

The LSPP also requires that there be only a 1 in 100,000 chance that multiple guesses within a minute will succeed. The TOE allows only interactive logins. If we assume that a human attacker could type quickly enough, including use of the secure attention key and entering a different password each time, to make a login attempt every two seconds, the chance of a match on a password guess within the minute would be1 in 103,000. Also this makes the unlikely assumption that the administrator has set the allowed number of failed login attempts to be greater than 30 – a more normal value would be 5.

In addition, the TOE implements the following aspects of the password mechanism that are stronger than what is implemented by many other systems (and add strength beyond the SOF-medium documented in the LSPP):

- An administrator can specify a minimum password length greater than 6 (and up to 16);
- An administrator can specify that passwords contain mixed case;
- An administrator can specify that passwords contain at least one non-alphabetic character (the alphabet for passwords can be as large as 124 character, including non-printing characters);

- An administrator can specify that passwords not found in a dictionary;
- Only N incorrect passwords can be tried on an account before the terminal is locked, where N has been configured by an administrator [this prevents repeated guessing ad infinitum];
- the encrypted passwords can not be viewed by untrusted users nor programs [therefore an attacker can not attempt to get passwords by cracking the encryption];
- Passwords are not ever requested across dial-up lines or networks and are never accepted by the TOE from such media/devices, and terminals and their communication lines must be physically protected [therefore an attacker has no opportunity to “sniff” passwords off communication lines]; Passwords are only ever requested or accepted after invocation of the trusted path mechanism [therefore an attacker can not spoof a request for a password];
- An administrator can configure the mechanism such that the last N passwords can not be reused;
- Passwords can not be used after their life time, which has been set by an administrator;
- Passwords must be changed after their expiration time, which has been set by an administrator;
- An administrator can specify that passwords must be changed after the initial login on an account.

The total protection and flexibility afforded by the features mentioned above qualifies the strength of secrets mechanism (FIA_SOS, IDNAUT) as SOF-high.

8.6 Rationale for Inclusion of Hardware in the TOE

All required hardware/firmware components are included in the TOE because: the LSPP suggests that a conformant TOE will include hardware; the hardware plays a role in implementing the following SRFs: FDP_RIP.2, FPT_RVM.1, FPS_SEP.1, and FPT_STM.1; and estimation of covert channel capacities requires making measurements on real hardware.

The following assumptions about the hardware/firmware base are made by the software:

I. The BIOS must:

a) Set hardware/firmware/microcode configuration settings as specified by the operator (they are specified when the TOE is not executing);

b) Take control of the system when the system is physically powered up or reset;

c) Run a set of hardware/firmware/microcode diagnostics during boot;

d) Transfer control to the TOE bootstrap loader, during boot, and then never execute again, unless explicitly invoked by the TOE, until another boot occurs;

e) Refrain from modifying the TOE bootstrap loader as it is copied from disk to memory;

f) Refrain from modifying any content of any disk or tape attached to the system.

II. Hardware_Domains: The hardware will provide a privilege domain mechanism that allows software executing in a privileged domain to be inaccessible to software executing in an unprivileged domain and that allows privileged software to restrict access to I/O instructions to privileged software. There will be at least two privilege levels. Hardware descriptors will control which domains can access which areas in physical memory and will control where unprivileged software can transfer into privileged software. These descriptors will only be modifiable by hardware-privileged software. Unprivileged software will have no way to disable or circumvent the descriptors and will have no way to gain privilege, other than by making controlled transfers to privileged software through the aforementioned descriptors. Unprivileged software will also have no way to modify other hardware descriptors, the mode of operation of the hardware, nor the control and configuration registers or the hardware.

III. Hardware_Faults: The hardware will provide a fault mechanism that allows privileged software to be notified of an attempt by unprivileged software to access memory outside its domain, beyond the limits of its own domain, never mapped, previously mapped, or mapped read-only. The information provided during the fault will allow privileged software to discern which case occurred and at what point in the execution of the unprivileged software. The software invoked at the time of the fault will be dictated by the settings in hardware descriptors. These hardware descriptors will be modifiable only by privileged software and unprivileged software will have no way to disable, circumvent, or corrupt the fault mechanism.

IV. Hardware_Integrity: The hardware/firmware/microcode components that handle data must maintain the integrity of that data and must maintain logical separation between separate logical requests.

V. Hardware_Virtual_Mem: The hardware must provide a virtual memory mechanism that allows the set of physical “pages” accessible to the current execution thread to be limited to a subset of all physical pages available to the current domain. The virtual-to¬physical mapping will also allow certain areas to be marked read-only. The virtual-to-physical mapping must be switchable only by hardware-privileged software and unprivileged software must have no way to disable, circumvent, or modify the mapping.

VI. Optional_HW: Hardware components other than those listed in chapter 2 can be added on to an XTS-400 system, but will be usable by the TOE and its users only if:

a) they are configured to answer to different I/O ports, memory addresses, and SCSI IDs than the listed components;

b) they are not using any bus resources of the system;

c) they do not interrupt the CPU;

d) they do not communicate with any of the required and optional components.

VII. Restrict_HW_Feat: The required and optional hardware/firmware/microcode is only allowed to contain features (such as vendor-specific extensions) beyond the listed standards, if one of the following:

a) The feature is inactive until explicitly invoked by software and is invoked only by special, obvious I/O or SCSI commands; or

b) The feature is inactive until explicitly invoked by physical user intervention and such user intervention is prevented by site physical or procedural controls; or

c) The feature is disabled by the standard TOE configuration or by site-specific procedures; or

d) Use of the feature does not: 1) allow display, retrieval or modification of information from another process nor from a global resource; 2) consume a global resource; 3) increase the performance of the system.

Section 9 Appendix A - List of Subjects

9.1 Subjects

XTS-400, Version 6.4(UKE) supports only one type of subject: a process[1].

9.1.1 Subject Attributes

For each subject, in addition to the current ring of execution, the system maintains the following security-relevant information in the kernel Active Process Table:

- The real user and group identifiers (IDs). This identifies the user and group responsible for the subject.
- The effective user and group IDs. This identifies the user and group on whose behalf the subject is operating. It is the effective user and group IDs that are used in the discretionary access checks.
- The clearance (MAC and MIC) of the user on whose behalf the subject is operating.
- The Mandatory Access Control (MAC) label (i.e., sensitivity label) of the subject.
- The Mandatory Integrity Control (MIC) label (i.e., integrity label) of the subject.
- The effective privileges of the subject.
- The maximum privileges of the subject.

9.1.2 Subject Creation and Destruction

Subjects can only be created by other subjects[2]. At the TSF interface, subjects are created via the *xts_load_process* and *fork* system calls. When *xts_load_process*, is used, the TSF creates a new subject environment that executes within the TSF until the TSF program loader relinquishes control. With *xts_load_process*, the new process environment is determined solely by the attributes of the program file. If the program being loaded has an integrity level greater than or equal to operator, the subject continues to execute within the TSF when loading is complete (i.e., it is a "trusted" subject). If the program is of an integrity level below operator, the subject executes outside the TSF when loading is completed (i.e., it is an "untrusted" subject).

Fork, the other subject creation path provided to untrusted code, creates a new subject whose environment is identical to that of the parent subject. In this case, the new subject executes within

[1] The definition is independent of the domain of execution of the process, which may change during the process' lifetime (for example, as the process transitions between hardware rings). In XTS-400 Version 6.4(UKE), as long as the process is active, it is a subject.
[2] The first subject in the system (the system loader) is created by the bootstrap loader.

the TSF until *fork* terminates; execution then resumes at the same point in parent and child. In all cases, the TSF interface provides trusted subjects with the ability to create both trusted and untrusted subjects; however, it restricts untrusted subjects to the creation of untrusted subjects. Subjects are destroyed by the *xts_release_process* system call.

Section 10 Appendix B – List of Objects

10.1 Information and Objects

Subjects provide no utility without information with which to work. This information is organized as "objects". Control over the interaction between subjects and objects is defined by the system security policy, which has mandatory and discretionary components. The discretionary controls are applicable only to the subset of objects that can be named by users outside the TSF. Supplemental mechanisms within the system provide finer control on the accessibility of information within objects, as well as providing assurance that reuse of objects cannot result in information flow.

The following sections describe the objects of the TOE. For each type of object, a description of the security-relevant characteristics of the object, as well as the methods of object creation and destruction, are given. The objects to be discussed are: processes, devices, semaphores, sockets[3], file system objects (files, directories, device special files, symbolic links, and named First-In First-Outs (FIFOs)), and memory objects (shared memory and unnamed pipes).

10.1.1 Semaphores

Semaphores, or more accurately semaphore sets, are used to coordinate access to resources. A semaphore set consists of one or more individual semaphores. Unlike file system objects and devices, semaphores are not included in the process address space. Semaphore sets contain the following security-relevant characteristics:

- The MAC label for the semaphore set, which cannot be changed once created.
- The creator of the semaphore set (user and group identifiers). This is set to the user and group identifiers of the process creating the semaphore set and, unlike the owner, it cannot be changed.
- The owner of the semaphore set (user and group identifiers).
- The access control list for the semaphore set.
- The ring identifier of the semaphore set (combined read/write).

10.1.2 Processes

Processes differ from file system objects in that they are considered to be both active subjects and storage objects. The storage object determination arises from the fact that (a) processes may be the target of an interprocess communication (IPC) message, and (b) processes contain accessible status information.

When viewed as an object, processes have the following security relevant characteristics:

[3] Sockets are protected resources, not named objects as the other objects discussed are.

- The MAC label of the process
- The effective owner of the process (user and group identifiers), for Discretionary Access Control (DAC) purposes
- The access control list for the process.

10.1.3 Devices

Devices are classified as objects in XTS-400, Version 6.4(UKE) because (a) devices serve as gateways for information, and, as such, may be viewed as abstract repositories of information; and (b) devices contain accessible status information. They have the following security-relevant characteristics:

- The MAC label of the device. This MAC label is constrained by the MAC label range associated with the system limits.
- The effective owner of the device (user and group identifiers), for DAC purposes
- The access control list for the device
- The device class (TRUSTED, USER, MOUNTED, and TERMINAL). The TRUSTED class is used for disk logical devices that are to be used by trusted services (e.g., check, fscheck, mkfsys). The USER class is used to specify those devices to which untrusted programs are to have access. The MOUNTED class is used to designate disk logical devices that are currently mounted. The TERMINAL class is used to designate devices that are configured for use as a terminal.

For disk devices, each partition has its own class (i.e., is a logical device). In order to change the class, all use of the device must be terminated. It must be unmapped (if a USER device) and have all file systems unmounted (if MOUNTED).

The kernel restricts device creation to subjects that possess an integrity level of administrator or greater. The kernel allows sharing of devices; however, devices can only be shared by processes running at the same MAC label as the device.

Additional information about and the amount of free space available on a disk holding a mounted kernel file system is returned if the user has MAC read access to the level of the device.

10.1.4 Sockets

Sockets (network and Unix-domain) are classified as objects in XTS-400, Version 6.4(UKE) because they are channels through which data can flow out of and into a process. Sockets are a little different than other TSF objects in the following ways: the object data may not be stored on the system, there may be no static “name” for the container of the data, data written cannot be read back, and specific “records” cannot be read. Instead, the object data is a dynamic set of packets to which there is no beginning, ending, or maximum size. The name of a network socket is also dynamic in that it can change according to what the sending and receiving processes agree upon. Like pseudo-terminals and unnamed pipes, sockets cease to exist one all processes have closed them.

The MAC and MIC label of the socket is set to the level of the creating process. For network sockets, this will be equal to the MAC and MIC label of the Ethernet device and of the TCP/IP daemon process (identical). If the Ethernet device were to change MAC levels, the socket would cease to operate. Sockets can only be used by processes at the same level, even if the processes are privileged.

DAC policy is not applied to sockets. Each socket can bind to a local port or identifier suitable for the protocol that is being used. TCP and UDP ports that are less than 1024 are considered privileged ports and can only be bound by the owner of the TCP/IP daemon.

The TSF ensures that only the creator of a socket can attach to a socket. The creating process must be at the same MAC label as the socket or Ethernet device.

10.1.5 File System Objects

File System objects are the basic information repository in the XTS-400, Version 6.4(UKE) system. File System objects possess the following security-relevant characteristics:

- The MAC and MIC label for the object.
- The owner of the object (user and group identifiers).
- The permissions and access control list for the object. This space is allocated by the kernel as part of the object for use by the DAC enforcement mechanisms in TSS.
- XTS-400, Version 6.4(UKE) uses file system objects to create a hierarchically structured file system. There are six types of file system objects: files, directories, device special files, symbolic links, Unix-domain sockets, and named FIFOs (pipes).

10.1.5.1 Files

The most common type of file system object in XTS-400, Version 6.4(UKE) is the regular file. The file header for an executable file also contains the following additional security-relevant information:

- The maximum authorised privilege set to be used
- The minimum integrity label required to execute the file.

10.1.5.2 Directories

A directory is a special type of file system object that is used to construct a hierarchical file system. The file system code in TSS enforces the restriction that, within a file system hierarchy, the mandatory level of a directory must be dominated by the mandatory level of each object in the directory.

The file system also supports a special type of directory called a deflection directory. This type of directory automatically redirects unprivileged users[4] into the appropriate hidden single level subdirectory.

10.1.5.3 Device Special Files

Device special files serve as the way that devices are designated through the file system. A device special file has a fixed structure: it is a UNIX-style major/minor number that is mapped by TSS into a device-unique ID for the actual device.

TSS does not allow a subject any direct access to a device special file. The only way to obtain information from a device special file is via the stat structure, which contains the major/minor numbers. It is up to untrusted software to map these major/minor numbers to device-unique IDs. Only when the actual device is opened via *xts_open_device or open* are checks made. The checks are based on information on the device, not on the device special file.

10.1.5.4 Named FIFOs

Named FIFOs provide a method to support the UNIX “named pipe” concept. They provide a permanent “First-In First-Out” communication path between multiple processes, at the request of the processes, within the bounds of the system security policy. Named FIFOs support multiple readers and multiple writer; locks are used to serialize access. DAC is enforced through the same mechanisms used for files.

10.1.5.5 Unix-Domain Socket File System Objects

Unix-Domain socket file system objects are simply a handle for getting to Unix-domain sockets (mentioned above). They are created by a “bind” operation and must be explicitly removed from the system when no longer wanted (i.e., they are not implicitly removed when all processes close the socket). Their security attributes can be changed, but a process attempting a “connect” must be at the level of the socket (and may need privilege to access the file system object). DAC policy is enforced during “connect” requests. Normal opens are not allowed against this kind of file system object.

10.1.5.6 Symbolic Links

Symbolic links appear as names in the file system, but simply “point” at another file system object. When an application performs a normal file system operation on a symbolic link, the operation is actually performed on the target of the link. If a symbolic link appears in a pathname, the target of the link is used in the path. There are special system calls to read the content, or get the status, of a symbolic link. Unlike “hard” links, the target object can be on another file system, can be downgraded with respect to the link, or can be non-existent.

[4] Users running at an integrity level above OSS can optionally bypass the deflection.

10.1.6 Memory Objects

Memory objects do not have a presence in the file system and are not persistent across system re-boots. They exist solely in system memory, except when portions of them may be swapped to disk. The TSF Kernel implements an internal memory object with full MAC, MIC, and DAC attributes, but this object is not directly visible outside the TSF. However shared memory and unnamed pipe objects are visible outside the TSF and are built on top of the internal, Kernel memory objects.

10.1.6.1 Shared Memory

Shared memory objects follow the Unix System V shared memory model. They can be shared between processes with a key or ID, or they can be private to the creating process. They persist until the system shuts down or until they are explicitly removed by the creator or owner. Shared memory is “mapped” into the address space of a process. The MAC and MIC level of shared memory objects can not be changed and is set to the level of the creating process by default.

10.1.6.2 Unnamed Pipes

Unnamed pipes serve as a data transfer conduit between processes. They have no inter-process “handle” and can only be shared between predecessor and ancestor in a “fork” relationship. An unnamed pipe ceases to exist as soon as the last process using it either closes the pipe or exits. Pipes are at the MAC and MIC level of the creating process and this level can not be changed. The DAC policy applied to pipes is that it is readable and writeable by the owner only.

This page is intentionally left blank

Section 11 Appendix C – TSF Commands

Command	Security Requirement
audit(1T)	**audit** is restricted to users whose current integrity level is at least operator and whose current security level is at the system maximum. Furthermore, to execute the **display** or **remove** functions, the user's current level must be the system maximum security level and at least the administrator integrity level.
check(1T)	**check** is restricted to users whose session integrity level is at least operator. The user must be at the same security and integrity level as the logical device containing the file system to be checked. The user also must be at or above the maximum security level of the file system.
copy_dump(1T)	**copy_dump** is restricted to users whose session integrity level is at least operator. The user must be at the same security level as the logical device containing the system dump.
ctl(1T)	**ctl** is restricted to users whose current integrity level is at least administrator.
daemon_edit(1T)	The user must be at the system minimum security and maximum integrity levels.
Dev_edit(1T)	The user must be at the system minimum security level and maximum integrity level.
dump(1T)	**dump** is restricted to users whose current integrity level is at least operator. The user must also possess the **shutdown_allowed** capability, as defined in the User Access Authentication database.
frestore(1T)	**frestore** is restricted to users whose current integrity level is at least operator. The user must be at the same security and integrity level as the save device, and at or above the maximum security level of the file system objects on the save media. **fsave** records the maximum potential security level of the file system objects on the save media. This level is determined by the level of the user. **frestore** will not restore an existing object if it has a higher integrity level than the user.
fsave(1T)	**fsave** is restricted to users whose current integrity level is at least operator. The user must be at the same security and integrity level as the save device, and at or above the security level of the objects being saved. **fsave** records the maximum potential security level of the file system objects on the save media. This level is determined by the level of the user. In the case of saving to removable media, the operator must attach an external label to the save media indicating the sensitivity of the saved information. In the case of a multivolume save, the operator should appropriately label all volumes and include sequence numbers in the labels.

Command	Security Requirement
fscheck(1T)	**fscheck** is restricted to users whose current integrity level is at least operator. The user must be at the same security and integrity level as the logical device containing the file system. The user also must be at or above the maximum security level of the file system.
fstab_edit(1T)	Can only be used by an operator or administrator.
ga_edit(1T)	The user must be at the system maximum security and maximum integrity level.
install(1T)	**install** is restricted to users whose integrity is at least administrator. The user must be at the same security and integrity level as the install device and at or above the `max` security level of the file system objects on the release media. **Install** will not set the discretionary access information of an existing object, except subtype. **Install** will set the mandatory access information of an existing object if it differs from that found on the release media.
mkfsys(1T)	**mkfsys** is restricted to users whose current integrity level is at least operator. The user must be at the same security and integrity level as the logical device on which the file system is being created, and at or above the maximum security level of the file system.
mount(1T)	**mount** is restricted to users whose current integrity level is at least operator. The user's current security level must be at or above that of the device and of the file system minimum security level.
partition_edit(1T)	The user must be at the system minimum security and maximum integrity level.
param_edit(1T)	The user must be at the system minimum security and maximum integrity level.
pq_edit(1T)	**pq_edit** is restricted to users whose current integrity level is at least operator and whose current security level is at system maximum.
Proc_edit(1T)	The user must be at the system maximum integrity level. In addition, if the user's security level is below the system maximum, only processes with the same or lower security classification and the same categories or a subset of those categories will be visible.
reboot(1T)	**reboot** is restricted to users whose current integrity level is at least operator. To use this command, the user must possess the **shutdown_allowed** capability, as defined in the User Access Authentication database.
sda(1T)	**sda** is restricted to users whose current integrity level is at least operator. The user must be at or above the (old) security level of the logical device.
sdc(1T)	**sdc** is restricted to users whose current integrity level is at least operator. The user must be at or above the security level of the logical device.
shutdown(1T)	**shutdown** is restricted to users whose current integrity level is at least operator. To use this command, the user must possess the **shutdown_allowed** capability, as defined in the User Access Authentication database.

Command	Security Requirement
sm_edit(1T)	The user must be at the system minimum security and maximum integrity level.
st(1T)	**st** is restricted to users whose current integrity level is at least operator.
start_daemon(1T)	The user must be at least operator integrity.
startup(1T)	**startup** is restricted to users whose current integrity level is at least operator.
stop_daemon(1T)	The user must be at least operator integrity.
tcpip_edit(1T)	The user must be at the system minimum security level and maximum integrity level.
tdc(1T)	**tdc** is restricted to users whose integrity is at least operator. The user must be at the same security and integrity level as the specified device.
Ua_edit(1T)	The user must be at the system maximum security and maximum integrity level.
unmount(1T)	**unmount** is restricted to users whose current integrity level is at least operator. The user's current security level must be at or above that of the device and of the file system minimum security level.

Command	Security Requirement
tp_edit(1T)	The user must be at the system maximum integrity level. The user must also be at the same or higher security level as any input file (for the **add** and **change** requests). For the **display** request, the user must be at the same or higher security level as the requested program file to display its attributes. For the **change** request, the user must be at the same or higher security level as the requested program file to change its attributes. **Warning:** Only untrusted programs may be installed without modifying the TCF and affecting any evaluation or certification rating that has been applied to the system. See Installing OSS Domain Programs in Chapter 1 of the *Trusted Programmer's Reference Manual*. **Warning:** Modifying or removing STOP programs in the **`/trusted`** directory or the **`/system`** directory could affect any evaluation or accreditation rating that has been applied to the system. **Warning:** Modifying or removing STOP programs in the **`/trusted`** directory or the **`/system`** directory can result in making the system unusable. The following special privileges are supported: **set_level** The ability to modify the mandatory security attributes of an object (security and integrity level). **upgrade_level** The ability to upgrade the mandatory security attributes of an object (security and integrity level). **set_discretionary_access** The ability to modify the discretionary security attributes of an object (access control information). **set_owner_group** The ability to change the owner and group associated with an object (for processes, the ability to change the real owner/group identifiers). **set_process_attributes** The ability to change restricted status information on a process (i.e., clearance level and process family identifier). **set_subtype_access** The ability to modify the object subtypes to which a process has access. **terminal_lock** The ability to lock and unlock terminals. **device_control_exempt** The ability to obtain control access to a device (i.e., the ability to issue privileged control functions). **simple_security_exempt** The ability to bypass the simple security property check (i.e., allows read up). **security_star_property_exempt** The ability to bypass the *-property security check (i.e., allows write down). **simple_integrity_exempt** The ability to bypass the simple integrity property check (i.e., allows read down).

Section 12 Appendix D – User Commands

Command	Security Requirement
ccp(1T)	The **ccp** command requires that the user have the **run_allowed** capability, as defined in the User Access Authentication database.
cdl(1T)	The **cdl** requires that the user have the **sl_allowed** capability, as defined in the Access Authentication database.
chd(1T)	none
cup(1T)	The **cup** command requires that the user have the **cup_allowed** capability, as defined in the Access Authentication database. Administrators attempting to change the password of another user must have additional capabilities.
df(1T)	The **df** command only displays information on mounted file systems whose maximum security level is less than or equal to the user's current security level.
disconnect(1T)	The **disconnect** command requires that the user have the **disconnect_allowed** capability, as defined in the User Access Authentication database.
fsm(1T)	For the following discussion, the following permissions are defined, depending on the current integrity level of the requester: **USER PERMISSIONS** `READ` The requester must be at the same security level or higher and at the same integrity level or lower as the object, must have discretionary read permissions to the object, and must have subtype access to the object. `WRITE` The requester must be at the same security level and integrity level as the object, must have discretionary write permissions to the object, and must have subtype access to the object. `MODIFY` The requester must be at the same security level and at the same integrity level as the containing directory, must have discretionary write and search permissions to the containing directory, and must have subtype access to the directory. **OPERATOR PERMISSIONS** `READ` The requester must be at the same security level or higher as the object, must have discretionary read permission to the object, and must have subtype access to the object. `WRITE` The requester must be at the same security level and at the same or higher integrity level as the object, must have discretionary write permissions to the object, and must have

Command	Security Requirement
	subtype access to the object. `MODIFY` The requester must be at the same security level and at the same or higher integrity level as the containing directory, must have discretionary write and search permissions to the containing directory, and must have subtype access to the directory. **ADMINISTRATOR PERMISSIONS** `READ` The requester must be at the same security level or higher as the object and must have subtype access to the object. `WRITE` The requester must be at the same security level, at the same or higher integrity level as the object, and must have subtype access to the object. `MODIFY` The requester must be at the same security level, at the same or higher integrity level as the containing directory, and must have subtype access to the directory. For the **change** and **delete** requests, the user must be at the same security level and integrity level as the requested object. In addition, the user must be the owner of the object or be executing as an administrator. An operator or an administrator may be at a higher integrity level than the requested object. With regard to the **change** request, only an administrator can turn on the *set-user-id* and *set-group-id* bits in the discretionary permissions. For the **cd** request, the user must have READ permissions to the directory and either must have discretionary execute (search) permission to the directory or be executing as an administrator. For the **copy** request, the user must have READ permission to the input file. When the target file exists, the user must have WRITE permission to the target file. When the target file is to be created, the user must have MODIFY permission to the directory to contain the new file. For the **create**, **link**, **mkdev**, **mkdir**, and **mkfifo** requests, the user must have MODIFY permission to the directory to contain the new object. In addition, to create a hard link to a file, the user must have discretionary execute (search) permission to the target file's containing directory and be at the same security level and same integrity level (or higher if running at operator or above) as the target. For the **deflect** and the **dispdef** requests, the user's current integrity level must be at least administrator. For the **display**, **dump** and **status** requests, the user must have READ permission to the requested object. For the **pwd** request, the user must have READ permission to the working directory. For the **rename** request, the user must have MODIFY permission to the old

Command	Security Requirement
	and new directories containing the object. In addition, if the old and new directories are different, the user must be at the same security level as the source object. If the object to be renamed is itself a directory, the user must also have READ permission to the object. If a display is to be sent to a system printer, the user's security level must be within the security level range of some printer with the specified printer class. When modifying the discretionary access of a file, if the new discretionary access has either the *set-user-id* or the *set-groupid* flag set, the following checks are made: ▪ If the effective group id of the **fsm** user does not match the new group id in the discretionary access, the *set-group-id* flag is reset. ▪ If user/group ownership of the file is to be changed, the *set-user-id* and *set-group-id* flags are reset. ▪ If the new discretionary access modes contain any write permissions, **fsm** rejects the request.
ifconfig(1T)	The user must be at the same mandatory and integrity level as the network for which information is to be displayed.
ikill(1T)	The **ikill** command requires that the user have the **kill_allowed** capability, as defined in the Access Authentication database.
kill(1T)	The **kill** command requires that the user have the **kill_allowed** capability, as defined in the Access Authentication database.
logout(1T)	none
reattach(1T)	The user must still have the clearance and group membership to match those of the process family to which to reattach. If the user's current access level and group membership are not the same as the process family's, the user must also possess appropriate capabilities.
run(1T)	The **run** command requires that the user have the **run_allowed** capability, as defined in the User Access Authentication database. The **run** command requires that the current integrity classification be 3 or lower.
scrlck(1T)	none
session(1T)	none
sg(1T)	The **sg** command requires that the user have the **sg_allowed** capability, as defined in the Access Authentication database.
sl(1T)	The **sl** command requires that the user have the **set_level_allowed** capability, as defined in the Access Authentication database.
sit(1T)	If new TSF file integrity checksums are to be generated, the user's integrity level must be at least "admin".
system(1T)	none

This page is intentionally left blank

Section 13 Appendix E – Auditable Events

What follows are the kinds of events that must be audited in order to meet the "basic" level of auditing, given the functional requirements included in the TOE.

Requirement	Audit events prompted by requirement
Audit Data Generation (FAU_GEN.1)	login/logout, start up/shutdown, requests to change user security attributes
User Identity Association (FAU_GEN.2)	login/logout
Audit Review (FAU_SAR.1)	Reading of information from the audit records (which can be attained by auditing any open of an audit file).
Potential Violation Analysis (FAU_SAA.1)	Unsuccessful login attempts.
Restricted Audit Review (FAU_SAR.2)	Unsuccessful attempts to read information from the audit records (which can be attained by auditing failed attempts to open an audit file for read).
Selectable Audit Review (FAU_SAR.3)	(none)
Selective Audit (FAU_SEL.1)	All modifications to the audit configuration that occur while the audit collection functions are operating.
Guarantees of Audit Data Availability (FAU_STG.2)	(none)
Action in Case of Possible Audit Data Loss (FAU_STG.3)	Actions taken due to exceeding a thresh-hold.
Prevention of Audit Data Loss (FAU_STG.4)	Actions taken due to the audit storage failure.
Complete Access Control (FDP_ACC.2)	(none)
Access Control Functions (FDP_ACF.1)	All requests to perform an operation on an object covered by the SFP.
Export of User Data Without Security Attributes (FDP_ETC.1)	All attempts to export information (which can be attained by auditing the open of any device and the running of the trusted mount, unmount, and fsave commands).
Export of User Data With Security Attributes (FDP_ETC.2)	All attempts to export information and to override human-readable output marking (which can be attained by auditing the open of any device, the request to override markings, and the running of the trusted mount, unmount, and fsave commands).

Requirement	Audit events prompted by requirement
Complete Information flow control (for Mandatory Access Control Policy) (FDP_IFC.2(1))	(none)
Complete Information flow control (for Mandatory Integrity Control Policy) (FDP_IFC.2(2))	(none)
Hierarchical Security Attributes (for Mandatory Access Control) (FDP_IFF.2)	All decisions on requests for information flow.
Hierarchical Security Attributes (for Mandatory Integrity Control) (FDP_IFF.2)	All decisions on requests for information flow.
Import of User Data Without Security Attributes (FDP_ITC.1)	All attempts to import user data, including any security attributes (which can be attained by auditing the open of any device and the running of the trusted mount, unmount, and frestore commands).
Import of User Data With Security Attributes (FDP_ITC.2)	All attempts to import user data, including any security attributes.
Subset Residual Information Protection (FDP_RIP.1)	(none)
Full Residual Information Protection (FDP_RIP.2)	(none)
Authentication Failure Handling (FIA_AFL.1)	Reaching of the threshold of allowed login attempts and attempts to restore system to ready state.
User Attribute Definition (FIA_ATD.1)	(none)
Verification of Secrets (FIA_SOS.1)	Rejection or acceptance by the TSF of any tested secret.
User Authentication Before Any Action (FIA_UAU.2)	All use of the authentication mechanism.
Single-use Authentication Mechanisms (FIA_UAU.4)	Attempts to reuse authentication details.
Protected Authentication Feedback (FIA_UAU.7)	(none)
User identification before any action (FIA_UID.2)	All use of the user identification mechanism, including the identity provided during successful attempts.

Requirement	Audit events prompted by requirement
User-subject binding (FIA_USB.2)	Success and failure of binding user security attributes to a subject (e.g., success and failure to create a subject). [This can be attained by auditing successful and failing login attempts, since the TOE will reject the login if a subject can not properly be created.]
Management of Security Functions Behavior (FMT_MOF.1)	All modifications in the behavior of the functions in the TSF.
Management of Security Attributes (FMT_MSA.1)	All modifications of the values of security attributes.
Static Attributes Initialization (FMT_MSA.3)	Modifications of the default setting of permissive or restrictive rules. All modifications of the initial values of security attributes.
Management of TSF Data (FMT_MTD.1)	All modifications of the values of TSF data, including audit data.
Management of Limits on TSF Data (FMT_MTD.2)	All modifications to the limits on TSF data and actions to be taken in case of violation of the limits.
Revocation (to authorized administrators)(FMT_REV.1(1))	All attempts to revoke security attributes and all modifications to the values of TSF data (this can be attained by auditing the use of trusted commands, auditing modification actions performed within trusted commands, and auditing security attribute changes to objects).
Revocation (to owners and authorized administrators) (FMT_REV.1(2))	All attempts to revoke security attributes.
Time-Limited Authorization (FMT_SAE.1)	Specification of the expiration time for an attribute. Action taken due to attribute expiration.
Management Operations (FMT_SMF.1)	Execution of operator and administrator commands. Additional auditing of trusted editor operations.

Requirement	Audit events prompted by requirement
Security Roles (FMT_SMR.1)	Modifications to the group of users that are part of a role and every use of the rights of a role (this can be attained by auditing "edits" performed by the trusted ua_edit command and by auditing use of every trusted command).
Assuming Roles (FMT_SMR.3)	Explicit requests to assume a role. Use of any function restricted to an authorized administrator role (identified in FMT_SMR.1).
Abstract Machine Testing (FPT_AMT.1)	Execution of the tests of the hardware/firmware base and the results of the tests. Note that this testing may be part of TSF self-testing.
Recovery from Failure (FPT_RCV.1)	The fact that a failure or service discontinuity occurred. Resumption of the regular operation. Type of failure or service discontinuity.
Non-Bypassability of the TSF (FPT_RVM.1)	(none)
TSF Domain Separation (FPT_SEP.1)	(none)
Reliable Time Stamps (FPT_STM.1)	Changes to the time.
TSF Testing (FPT_TST.1)	Execution of the TSF self tests and the results of the tests.
TSF-initiated session locking (FTA_SSL.1)	Authentication failures during attempts to unlock.
User-initiated locking (FTA_SSL.2)	Authentication failures during attempts to unlock.
TSF-Initiated Termination (FTA_SSL.3)	Logout of an interactive.
Default TOE Access Banners (FTA_TAB.1)	(none)
TOE Access History (FTA_TAH.1)	(none)
TOE session establishment (FTA_TSE.1)	Login failure due to password expiration.
Trusted Path (FTP_TRP.1)	All attempted uses of the trusted path functions. Identification of the user associated with all trusted path failures, if available.

Section 14 Appendix F – Selectable Audit Events

What follows are the specific events that can be audited by the TOE. These are used to satisfy the "basic" level of auditing requirements shown in *Appendix F – Selectable Audit Events*, but go beyond those requirements.

The TOE shall be able to audit the following events:

system change event
branch block runout
device access change
device creation
device deletion
device owner change
device close
IPC message sent
mount space runout
process creation
process deletion
process owner change
process real user change
process subtypes change
FS object subtypes change
device subtype violation
FS object subtype violation
shared memory unmap
semaphore access denial
shared memory DAC
program loader failed
FS object access change
FS object close
FS object creation
FS object deletion
FS object open
FS object remove link
socket bind failure
socket accept
socket sendto recvfrom
file system access violation
data block runout
device access violation
device DAC violation
device open
device subtypes change
disk error
mount access violation
process access violation
process DAC violation
process fork
process privilege change
process space runout
FS object access violation
unmount busy file system
process subtype violation
shared memory map
semaphore object
semaphore access change
device class change
setuid program
FS object add link
FS object chdir
terminal lockout
FS object name change
FS object owner change
socket open close
socket connect
FIFO action
Internet inbound connect failure

ICMP redirect	admin command
cdl command	**ctl** command
cup command	device start error
fsm error	login
logout	operator command
print with no markings	**sg** command
shutdown command	**sl** command
st command	**startup** command
trusted editor request	UPS status change
sit command	memory object runout
crypto event	**entropy pool runout**
deny write	**integrity failure**
screen unlock authentication	**user lockout**
device open	**resource exhaustion**
ICMPv6 redirect	**sit verification failure**

Section 15 Appendix G -Acronyms

CASS	Commodity Application System Services
CC	Common Criteria
EAL	Evaluation Assurance Level
IT	Information Technology
OSS	Operating System Services
PP	Protection Profile
SF	Security Function
SFP	Security Function Policy
SOF	Strength of Function
ST	Security Target
TFM	Trusted Facility Manual
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSFI	TSF Interface
TSP	TOE Security Policy
TSS	TSF System Services

This page is intentionally left blank

Section 16 Appendix H – Functional Requirements Required by Existing XTS-400 Customers

The following functional requirements are not required by the PPs to which the TOE is conforming, but are required by existing TOE customers:

1. FAU_SAA.1
2. FMT_MSA.2
3. FPT_TST.1

This page is intentionally left blank