

# BAE SYSTEMS

## XTS-400™

# Trusted Computer System

## *Technical Overview*

### **ABSTRACT**

The BAE Systems Information Technology XTS-400™ Trusted Computer System is the successor to the worlds' only National Information Assurance Partnership (NIAP) evaluated high-assurance general-purpose computer system. The XTS-400 features Linux® compatibility and supports contemporary Intel® hardware speeds. It is designed to require no porting of common applications which run on Linux, to be easy to develop for and to allow the use of a wide variety of modern development tools. It is designed for and has been submitted for Common Criteria Evaluation against a robust Security Target at a high Evaluation Assurance Level. Its predecessor, the XTS-300™ has long provided defense, intelligence, diplomatic, law enforcement, and other security conscious communities with an extremely secure trusted system running on the latest generation of Intel server-class hardware. Each generation of the XTS family of systems has been successfully evaluated by the National Security Agency at a level requiring not only incorporation of required security features, but also a very high level of assurance and testing. This level of assurance allowed the XTS-300 to be easily accredited to handle data at a wide range of sensitivity (e.g., classification) levels in a wide range of operational environments. The XTS-300 is currently deployed in several accredited, operational trusted applications worldwide. The XTS-400 is designed to provide the same high level of security in many kinds of applications, including specialized applications such as network guards or filters for handling the semi- or fully automatic downgrading of information.

**BAE SYSTEMS INFORMATION TECHNOLOGY**  
*XTS-400™ Trusted Computer System Technical Overview*

Michael W. Focke

XTS Product Manager

BAE Systems Information Technology

2525 Network Place

Herndon, VA 20171 U.S.A.

TEL: (703)563-8087

FAX: (703)563-8013

[mike.focke@baesystems.com](mailto:mike.focke@baesystems.com)

This page intentionally left blank

## TABLE OF CONTENTS

Scope	Page
1 HISTORY.....	1
2 INTRODUCTION.....	3
3 SECURITY FEATURES .....	5
3.1 Significance of Common Criteria Evaluation Assurance Levels (EAL) .....	6
4 SYSTEM ARCHITECTURE.....	9
4.1 Domain 0: Security Kernel.....	9
4.2 Domain 1: Trusted System Services .....	9
4.3 Domain 2: Operating System Services (OSS).....	9
4.4 Domain 3: Application Domain .....	9
4.4.1 Trusted Software .....	9
4.4.2 Software Development Environment (SDE).....	9
4.5 Trusted Databases .....	10
5 PHILOSOPHY OF PROTECTION.....	13
5.1 Reference Monitor.....	13
5.2 TSF Assurance Mechanisms .....	13
5.2.1 Domain Isolation.....	13
5.2.2 Integrity .....	13
5.2.3 Process Isolation.....	13
5.2.4 Trusted Path.....	13
5.2.5 Subtypes .....	13
5.3 Mandatory Security Policy.....	14
5.4 Mandatory Integrity Policy .....	14
5.5 Discretionary Access Controls.....	14
5.6 Subtype Control.....	15
6 HARDWARE.....	17
7 XTS-300/XTS-400 APPLICATIONS .....	19
7.1 Guard-Enabling Technologies.....	19
7.1.1 DataSync Guard .....	19
7.1.2 Standard Automated Guard Environment.....	19
8 Why would you use the XTS-400? .....	21

## LIST OF FIGURES AND TABLES

Table 1. Common Criteria assurance levels and their requirements.....	8
Figure 1. STOP Four-Domain Architecture.....	11
Table 2. Hardware Details of the XTS-400 Model 2800 .....	17

This page intentionally left blank

# 1 HISTORY

The BAE Systems Information Technology XTS-300™ Trusted Computer System is the only National Security Agency (NSA) evaluated high-assurance general-purpose computer system. The XTS-300 has long provided defense, intelligence, diplomatic, law enforcement, and other security conscious communities with an extremely secure system running on the latest generation of Intel server-class hardware. Because it is based on commodity hardware, the XTS-300 was positioned to take advantage of the frequent hardware advances in the Intel x86 hardware base and in the SCSI subsystem.

Each new generation of the XTS family has been evaluated by the NSA, according to the National Security Agency's (NSA) Trusted Computer System Evaluation Criteria (TCSEC), at the Class B3 level, proof that the system provides a very high level of security functionality. The XTS-300 uses the Secure Trusted Operating Program (STOP™) operating system. STOP 5.2.E's entry in the NSA Evaluated Products List can be viewed on the Internet at: <http://www.radium.ncsc.mil/tpep/epl/entries/CSC-EPL-92-003-E.html>

The XTS-300 is used both by BAE - IT and by customers as a platform upon which to build applications which filter data and enforce security policies. Filtering is the process which allows the application of rules-based inspection and selection/rejection criteria. It allows the selected data to safely pass from one protected security classified level to another security classified level. Many of these programs have been called "Guards" because they guard one network from the other using the hardware enforced protections of the XTS while still allowing the flow of selected data through carefully architected and fully accredited logic paths between networks of differing levels.

The XTS-400 is a successor product, built on the hardware and software foundation established by the XTS-300. The XTS-400 uses the same hardware architecture and much of the same Operating System Kernel design and code as the XTS-300. The XTS-400 uses the same hardware and software protection mechanisms that have been repeatedly proven in TCSEC Evaluations. The XTS-400 adds support for contemporary hardware speeds and a robust set of Linux® Application Programming Interfaces (APIs) so that the thousands of programs and commands written for Linux can, in their binary form and without porting

or recompilation, be trivially copied to the XTS and run under the protection of the XTS's secure architecture (Application Binary Interface or ABI).

Programs on the XTS-400 which use the APIs unique to the XTS use the same security enforcing features proven on the XTS-300.

The operating system that runs on the XTS is called the Secure Trusted Operating Program (STOP™). STOP, in version 6 and later, supports Linux APIs as well as the security enforcing APIs and commands unique to the XTS.

STOP 6.0 received an EAL4 Augmented rating in March of 2004.

STOP 6.1.E received its evaluation at the EAL5 Augmented level in March 2005. This included penetration testing performed by the National Information Assurance Partnership (NIAP).

See <http://niap.nist.gov/cc-scheme/ValidatedProducts.html>.

This page intentionally left blank

## 2 INTRODUCTION

The XTS-400 provides multilevel secure Target of Evaluation (TOE) Security Functions (TSF) that allow simultaneous processing and storage of data at different classifications or sensitivities and needs-to-know (categories/compartments) by users with different clearances and needs-to-know. The XTS-400 can eliminate arbitrary over-classifying of data, as occurs in system-high mode.

The system's design for high robustness/risk environments implies not only incorporation of particular security features, but a very high level of assurance. This level of assurance allows the XTS-400 to be accredited to handle data at a wide range of sensitivity (e.g., classification levels) in a wide range of operational environments. Evaluation, certification and accreditation efforts have begun on applications that use the XTS-400 as a multi-level application platform (and many similar certifications and accreditations have been completed on the XTS-300). The XTS-400 is designed to provide a high level of security in many kinds of applications, including specialized applications such as network guards or filters for handling the semi- or fully automatic downgrading of information.

The XTS-400 is a general-purpose computing system in that it can be used for a wide range of applications— from multi-user workstation to trusted guard to trusted server. Its predecessor, the XTS-300 is currently deployed in fully accredited, operational systems worldwide, including the NSA's Defense Information Infrastructure (DII) Guard, the Defense Information Systems Agency's (DISA's) Command and Control Guard, the Federal Bureau of Investigation's "Cyclone" Guards, the State Department's Unclassified Telegram Guard Processor, the Air Force's F-22 Secure Interface System, the Department of Energy's FTP Guard, Novell Corporation's NICI Public Key Infrastructure, and trusted applications in the U.S. Intelligence Community, and in Canada's Department of Foreign Affairs, Industry, and Trade (DFAIT).

STOP 6 is currently supporting a guard application that is in operational test within a defense message handling system. This guard uses a mix of BAE - IT supplied middleware, COTS software and purpose built filters to achieve its mission.

This page intentionally left blank

### 3 SECURITY FEATURES

The XTS-400 product is a combination of STOP 6, a multilevel secure operating system, and a BAE - IT supplied Intel x86 hardware base. (STOP is not licensed to run on non-BAE - IT supplied hardware and would lose its evaluated status if this were attempted.)

The system provides mandatory access control that allows for both a security (MAC) and integrity (MIC) policy. The mandatory security policy enforced by the XTS-400 is based on the Bell and LaPadula security model. Beyond the historic requirements for a TCSEC Orange Book B3 system, the XTS-400 provides a mandatory integrity policy (which is required by draft Common Criteria medium- and high-robustness profiles), an extra subtype policy, and a familiar, Linux-like environment for single-level applications. Integrity can be used for, among other things, virus protection. The mandatory integrity policy is based on the Biba integrity model. The system also implements discretionary access control (DAC) and provides for user identification and authentication needed for user ID-based policy enforcement. The system also provides an additional policy mechanism, “subtypes,” which is not required by the TCSEC or Common Criteria and which can be used in a customer-specific way in conjunction with MAC, MIC, and DAC controls.

At every level of the system, and for each database, application, user, terminal, and process, there is a level of security.

The OS operates using a construct now referred by Intel to as “domains of isolation.” (These used to be called “Rings”.) Each domain is exclusive. Domain 0 — the Security Kernel with the system’s highest level of security — is inaccessible by users. It is within this domain that I/O device drivers reside, so that no one at any time can gain unauthorized access to device drivers. The OS is tamperproof due to the domain protection mechanism. Even processes are restricted by Domain privileges, allowed to send messages only to those other processes that have the same or lesser domain privileges. A terminal cannot simultaneously connect to processes at different MAC levels. To connect to the process with a different level, the user must first disconnect from the current process(es). All of these conventions are enforced within the system itself.

The XTS-400’s STOP operating system is a multiprogramming system that can support terminal connections for multiple users. Up to 200 processes can run concurrently, each with up to four gigabytes of virtual memory. STOP supports most of the Linux

interface for applications software (Application Programming Interface compatibility), and can run most object programs compiled on Linux with no change (Application Binary Interface compatibility). An X-Windows graphical user interface (GUI) is supported outside the TOE Security Functions (TSF), and is available at the console for work by untrusted users. All Windows on the display are at the same level (multi-level cut-and-paste is not supported). A trusted path mechanism is provided by the implementation of a Secure Attention Key (SAK). Initiation of the trusted path causes suspension of the GUI, and absolutely isolates the trusted command interface from the GUI environment.

Network connectivity is allowed in the evaluated configuration. TCP/IP and Ethernet (10BaseT/100BaseT) are built in to the TSF, while all network servers (e.g., SMTP) except ICMP are run outside of the TSF. Within an evaluated configuration, network attachments must be single-level, while multiple networks can each be at the same or a different level. Other operating systems have received evaluations, but lose their evaluated status when connected to a network or to networks of differing classifications. The XTS is different, its network access is designed with security as its highest priority and the XTS is specifically allowed to connect to multiple single level networks simultaneously while retaining its evaluated status

The separation of administrator and operator roles is enforced using the integrity policy. The system enforces the “principle of least privilege” (i.e., users should have no more authorization than that required to perform their functions) for administrator and operator roles. All actions performed by privileged (and normal) users can be audited. The audit log is protected from modification using integrity and subtype mechanisms.

STOP also provides an alarm mechanism to detect the accumulation of events that indicate an imminent violation of the security policy. Individual accountability is provided with an auditing capability. Data scavenging is prevented through object reuse (i.e., residual data) prevention mechanisms.

The TSF exhibits strong architectural characteristics: minimization, layering, abstraction, and data hiding. The TSF makes use of hardware features to provide process separation and TSF isolation and has been designed and implemented to resist penetration. All attempts by evaluators and certifiers to penetrate the TSF have failed. The system design is based on a written security model and other high-level design documentation.

The multilevel security features of the XTS-400's Target of Evaluation Security Functions (TSF) enforce trusted labeling, a mandatory access control policy, and a mandatory integrity control policy that enable the system to allow users with different clearances and needs-to-know to simultaneously store and process information that exists at different classification levels or sensitivities and/or in different need-to-know categories or compartments. Authorized users can process information on the XTS-400 at its actual sensitivity level, helping to eliminate the arbitrary over-classification of information that often occurs in system-high operations.

The XTS-400 is designed to meet Common Criteria EAL6 assurance requirements. For the most part, these EAL requirements are very similar to the Orange Book (i.e., TCSEC) B3 assurance requirements.

### 3.1 Significance of Common Criteria Evaluation Assurance Levels (EAL)

Traditionally, systems have been separated into low, medium and high assurance categories with NSA guidance provided as to the appropriateness of the solution to the task of separating networks and data of specific classifications. Thus a system rated High Assurance could connect to and separate from classifications which were more apart or were, themselves, higher in importance. The NCSC's *Guidance for Applying the DoD TCSEC in Specific Environments* (known as "The Yellow Book") included a "Security Index Matrix for Open Security Environments" (Open and Closed indicate the strictness of security controls in the system's development environment, not its operational environment). The matrix showed the minimum TCSEC rating a system should have to allow users within a given range of authorizations to access data within a given range of classifications. There does not appear to be anything similar yet issued which references the Common Criteria assurance levels.

*Table 1*, taken from the Common Criteria security assurance requirements document (CCIMB-99-033) provides a look at the areas that change as you move between assurance levels. Each of the Assurance Categories has varying requirements that get stricter as the numbers get higher. You can see from the table that there are 11 differences as you go from EAL4 to 5 and another 13 differences between EAL5 and 6.

#### EAL4

"EAL4 permits a developer to gain maximum assurance from positive security engineering based on good commercial development practices which, though rigorous, do not require substantial specialist knowledge, skills, and other resources. EAL4 is the highest level at which it is likely to be economically feasible to retrofit an existing product line."

"EAL4 is applicable to those circumstances where developers or users require a moderate to high level of independently assured security in conventional commodity (OSs) and are prepared to incur additional security-specific engineering costs."

Windows NT2000 and Trusted Solaris 8 are evaluated at EAL4. Windows NT was evaluated at only C-2 and Trusted Solaris at only B-1 under the TCSEC criteria. These might be thought of as Medium Assurance platforms. Each was evaluated against a non-demanding Protection Profile. Note that Windows does not provide mandatory access controls and Solaris does not provide mandatory integrity.

#### EAL5

"EAL5 permits a developer to gain maximum assurance from security engineering based on rigorous commercial development practices supported by moderate application of specialist security engineering techniques. Such a (OS) will likely be designed and developed with the intent of achieving EAL5 assurance."

"EAL5 is ... applicable in those circumstances where developers or users require a high level of independently assured security in a planned development and require a rigorous development".

EAL5 requires "semiformal design descriptions, the entire implementation, a more structured (and hence analyzable) architecture, covert channel analysis, and improved mechanisms that provide confidence that the (OS) will not be tampered with during development."

EAL5 begins the series of levels which require the OS developer to design for security first. They also require tests and documentation to be written to exacting standards. STOP 6 and the XTS-400 will be evaluated at EAL5 initially.

In general, you can be sure that at EAL5 and higher, NSA has had a direct hands-on role in the analysis and vulnerability testing.

There are currently no other Operating Systems in evaluation or Evaluated at EAL-5 except STOP.

### **EAL6**

“EAL6 permits developers to gain high assurance from application of security engineering techniques to a rigorous development environment in order to produce a premium (OS) for protecting high value assets against significant risks.”

“EAL6 is ... applicable to the development of security TOEs for application in high risk situations where the value of the protected assets justifies the additional costs.”

EAL EAL6 requires “more comprehensive analysis, a structured representation of the implementation, more architectural structuring (e.g. layering), more comprehensive independent vulnerability analysis, systematic covert channel identification and improved configuration management and development environment controls.”

EAL6 begins the high assurance levels which require enormous efforts in documentation and test writing involving significant time, expense and expertise to achieve. STOP 6 and the XTS-400 are designed to be able to achieve EAL6 evaluation.

### **Assurance Levels**

The more the system is asked to do, the more functionality must be defined in the selected Security Target or Protection Profile. Two products with similar EAL ratings might have vastly different security functionality depending on the environment they are asked to defend, the risks associated with that environment and thus the functionalities demanded in the Security Target or Protection Profile against which the product is evaluated.

The key requirement for the consumer of a Common Criteria product is to correctly define their environment and its associated risks. Then to then select an appropriate Security Target or Protection Profile. And then to select a product whose Evaluation Assurance Level (EAL level) is the one appropriate for the classification and sensitivity of data and the classification of the networks intersecting at the platform. Don't assume that, because two products have similar EAL levels, they offer the same protection.

One could have chosen a less inclusive Protection Profile, the other a much more inclusive one (i.e., one may meet many more security functional requirements than the other).

The XTS-400's Trusted Facilities manual, its Security Model and its Security Target are the best guidelines for understanding the functionality of the XTS-400's security features.

Assurance Class	Assurance Family	EAL1	EAL2	EAL3	EAL4	EAL5	EAL*	EAL6	EAL7	
Configuration Management	ACM_AUT				1	1	1	2	2	Automation of CM
	ACM_CAP	1	2	3	4	4	4	5	5	Prevents unauthorized modification
	ACM_SCP			1	2	3	2	3	3	Tracking changes and security flaws
Delivery and Operation	ADO_DEL		1	1	2	2	2	2	3	Detects modification during delivery
	ADO_IGS	1	1	1	1	1	1	1	1	Provide guidance for installation & set-up
Development	ADV_FSP	1	1	1	2	3	3	3	4	Functional Specification
	ADV_HLD		1	2	2	3	4	4	5	High level descriptions
	ADV_IMP				1	2	3	3	3	Explain functions and dependencies
	ADV_INT					1	2	2	3	Internal design requires modularity and layering
	ADV_LLD				1	1	2	2	2	Low level design detail
	ADV_RCR	1	1	1	1	2	2	2	3	Documentation matches
	ADV_SPM				1	3	1(2)*	3	3	Security policy modeling formality
Guidance Documents	AGD_ADM	1	1	1	1	1	1	1	1	Guidance for administration
	AGD_USR	1	1	1	1	1	1	1	1	Guidance for users
Life cycle Support	ALC_DVS			1	1	1	2	2	2	Controlled development process
	ALC_FLR						(3)*			Systematic flaw handling
	ALC_LCD				1	2	1	2	3	Must have life cycle model
	ALC_TAT				1	2	1	3	3	Basic requirements for development tools
Tests	ATE_COV		1	2	2	2	3	3	3	Testing philosophy & procedures
	ATE_DPT			1	1	2	2	2	3	High level design testing
	ATE_FUN		1	1	1	1	2	2	2	Functional testing
	ATE_IND	1	2	2	2	2	2	2	3	Independent Testing
Vulnerability Assessment	AVA_CCA					1	2	2	2	Covert channel analysis
	AVA_MSU			1	2	2	3	3	3	Misuse analysis
	AVA_SOF		1	1	1	1	1	1	1	Strength of function analysis
	AVA_VLA		1	1	2	3	4	4	4	Analysis for vulnerabilities

Table 1. Common Criteria assurance levels and their requirements

The \* indicates that NSA has defined a new requirement which is not included in the Common Criteria Security Assurance Requirements document.

## 4 SYSTEM ARCHITECTURE

The STOP operating system provides Trusted Security Functions (TSF), which enforce security policy, and untrusted commands, which generally provide the user interfaces familiar to Unix/Linux users.

The XTS-400's Pentium® CPU's four-domain chip architecture reinforces the STOP operating system's mandatory security and integrity access control policies by physically isolating security domains in hardware, preventing system processes from tampering with each other. The CPU domain architecture restricts access to segments, pages, and instructions. There are four levels: Level 0 to Level 3, with Level 0 being the most privileged level. The CPU also provides multiple checks for protection violations within memory references.

As illustrated in *Figure 1*, both non-TSF processes and TSF processes are mapped into the four-domain architecture in the same manner. Both types of processes map to the same Domain 0 Kernel, Domain 1 Trusted System Services, and Domain 2 Operating System Services. More information on the processes in each STOP domain follows.

### 4.1 Domain 0: Security Kernel

The most privileged domain, the Security Kernel contains the Reference Monitor that enforces system security policy. Small and well structured to enable complete security evaluation, testing, and verification, the Kernel provides basic OS services (resource management, process scheduling, interrupt and trap handling, auditing, mandatory and discretionary access policy enforcement for processes and device objects; I/O device drivers reside in Domain 0). Domain 0 processes cannot be called or modified by users.

### 4.2 Domain 1: Trusted System Services

TSS provides networking, I/O, file system management, and file system object discretionary access policy enforcement for both trusted and untrusted system processes and applications. The TSS environment is controlled by the Security Kernel, which enforces mandatory security, mandatory integrity, and subtype control on the TSS and all other XTS-400 operations. Domain 1 processes cannot be called or modified by users.

### 4.3 Domain 2: Operating System Services (OSS)

OSS provides the Linux APIs expected by applications written for Linux or using Linux tools. OSS also provides XTS-proprietary APIs to help manage and use the trusted aspects of the system. OSS then translates these APIs into trusted OS primitives provided by the Kernel and TSS. OSS also does some management of application signals and process groups. Applications can interface only with the OSS portion of the TSF – they cannot call TSS or the Kernel directly.

### 4.4 Domain 3: Application Domain

Both trusted and untrusted applications execute in Domain 3 (and can only execute in Domain 3). If a process is running at low integrity and has no privileges, it is considered untrusted. These untrusted applications include the user commands and tools that are familiar to Linux/Unix users.

#### 4.4.1 Trusted Software

Trusted Software includes all security-relevant functions that operate as independent services (e.g., the security map editor). Some Trusted Software functions may bypass the TSF's mandatory and/or discretionary controls, e.g., to enable high-integrity users to establish/modify the file system hierarchy to accommodate use of high-integrity nodes. Trusted Software functions are available to system operators and administrators for security-related housekeeping (user registration/removal, password assignment, system installation/configuration, and privileged tasks not supported by other STOP components).

A few Trusted Software functions, such as start-up of application sessions, are available to Domain 3 users. The user interfaces to Trusted Software supports many Linux/UNIX features while proprietary interfaces support the XTS-unique security enforcing interfaces to the operating system kernel.

#### 4.4.2 Software Development Environment (SDE)

The XTS-400 Software Development Environment (an optional software product) enables developers to write their own trusted and untrusted applications. Typically, "C" is the programming language though other languages and shells supported by Linux could be used.

## **4.5 Trusted Databases**

The XTS-400's trusted databases contain sensitive user and group access, session control, and print queue information protected from unauthorized modification by unprivileged processes. Trusted databases can be manipulated only by user-developed trusted processes, or trusted editors used by system/security administrators.

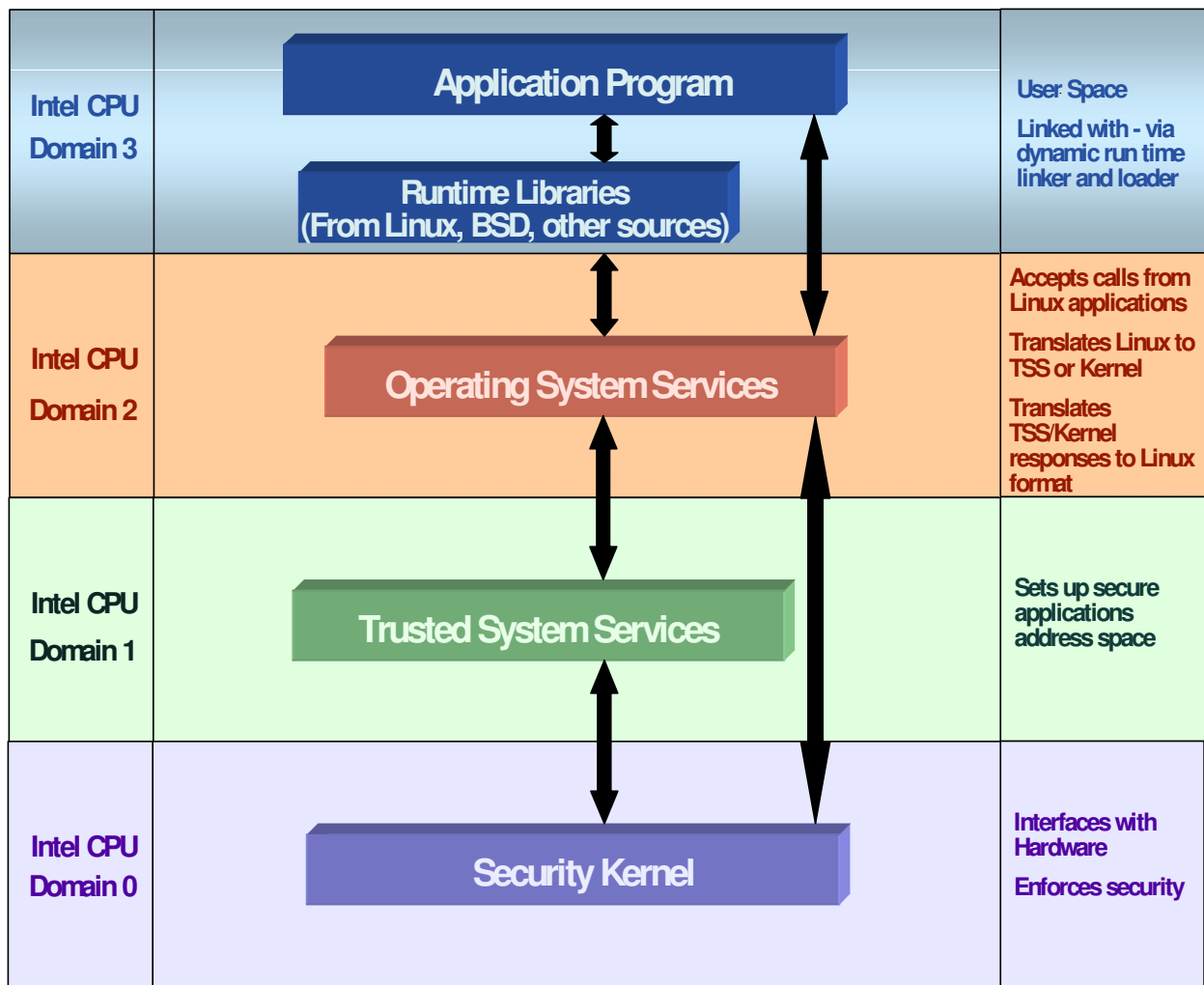


Figure 1. STOP Four-Domain Architecture

This page intentionally left blank

## 5 PHILOSOPHY OF PROTECTION

To enforce the mandatory access policies that make the XTS-400 multilevel secure, STOP implements a Reference Monitor. The Reference Monitor enforces authorized access relationships between system subjects (trusted and untrusted processes acting on a user's behalf to perform accesses) and system objects (file system objects, devices, semaphores, sockets, and processes).

Trusted subjects are used mainly for functions that manipulate the system's trusted databases or perform strictly controlled circumventions of the TSF's mandatory and/or discretionary access rules. A typical example of a trusted process is the Regrader (reclassifier/relabeller) process in a Trusted Guard. Except for those few processes that must update a trusted database or bypass STOP's access controls, untrusted subjects can be relied upon to perform most application functions.

### 5.1 Reference Monitor

The Reference Monitor compares each attempt by a subject to reference (access) an object against a list of reference types (read, write, and/or execute) the subject is authorized to perform on that object. The Reference Monitor's access validation mechanism is invoked for every reference by a subject to an object, thus preventing any unauthorized accesses. To ensure its integrity, the Reference Monitor's access control/validation mechanism is programmed to be tamperproof. The Reference Monitor is implemented in the TSF, which derives from the Intel CPU's four-Domain isolation mechanisms the absolute separation of the Reference Monitor from Domain 3 functions and applications running on the system.

### 5.2 TSF Assurance Mechanisms

All software processes on the XTS-400 are isolated from one another by the Security Kernel's enforcement of the Bell-LaPadula security and Biba integrity rules. Processes may access only information they dominate, and the entire TSF is protected from unauthorized tampering via the following mechanisms.

#### 5.2.1 Domain Isolation

Domain isolation protects code and data in the Kernel from modification by processes in any other Domain; protects the code and data in each Domain from

modification by users/processes in any less privileged Domain.

#### 5.2.2 Integrity

The system's mandatory integrity mechanism sets integrity levels of TSF program files, databases, and most trusted software processes to operator or higher; excludes untrusted users (subjects) from the TSF by limiting their maximum integrity to less than that of TSF objects.

#### 5.2.3 Process Isolation

Trusted Software processes (like most applications) keep their working data in process-local data areas that cannot be shared by other processes or accessed by untrusted software. The Kernel prevents any process from directly accessing another process' program text and local data and prevents untrusted processes from modifying trusted processes and their data.

#### 5.2.4 Trusted Path

Before a terminal can communicate with the TSF, the operator must press the Secure Attention Key (SAK) to disconnect the terminal from an untrusted process(es). This ensures that the user is communicating with the TSF, not with an untrusted process spoofing a TSF process. Any unlocked terminal used by trusted software is protected from untrusted software and other users' processes by a terminal-unique device subtype. When a user enters the TOE via the secure path, the secure server detaches the terminal's subtype from all untrusted processes associated with the session. Terminal access to untrusted processes is restored only after the user explicitly exits the trusted environment.

#### 5.2.5 Subtypes

Subtypes are used like tokens: to access an object on the system, a subject must possess the object subtype for that object. The system's subtype mechanism is used by the Kernel to restrict access to processes, trusted databases, and devices. As described above, the primary use of subtypes is to provide control over the Trusted Path: when the SAK is pressed, the Server changes the subtype of the terminal to prevent any untrusted process from accessing it. Subtypes are also used by the file system management (FSM) process to assure that FSM gets exclusive access to the file object. When it accesses the file object, FSM resets the subtype to one to which only FSM has access. After it finishes

processing the file, FSM resets the file to its original subtype. Finally, subtypes are used to protect the system's trusted databases, by giving only trusted programs the appropriate subtypes needed to access the databases.

### 5.3 Mandatory Security Policy

Each object in the XTS-400 is referenced by a unique identifier, and has its own set of access and status information (including subtypes) to implement non-hierarchical mandatory access controls based on need-to-know, and mandatory and discretionary access attributes. An object's mandatory access information includes its mandatory security and integrity levels and categories (or compartments); this information provides the basis on which the Kernel makes mandatory access control decisions related to the object.

Subjects in the XTS-400 can only reference objects according to the NCSC-approved Bell-LaPadula formal mathematical model of computer security policy (D. E. Bell and L.J. LaPadula. Secure computer systems: mathematical foundations. Technical Report ESDTR73278, MITRE Corp., Redford, MA, Nov 1973). This policy is implemented by a set of security rules designed to protect data from unauthorized access.

*Simple Security:* Subject may read or execute object only when subject's security level dominates objects.

*Security\*Property:* Subject may write object only when object's security level dominates subjects. XTS-400 security\*property allows subject to write object only when subject and object are at the same security level. This prevents lower level subjects from writing higher-level objects they cannot later access.

The system supports 16 hierarchical security classifications and 64 independent non-hierarchical "need-to-know" security categories/compartments.

### 5.4 Mandatory Integrity Policy

The XTS-400 enforces K. J. Biba's integrity policy (K. J. Biba. Integrity considerations for secure computer systems. Technical Report ESDTR76372, MITRE Corp., Redford, MA, 1977), a corollary to the Bell-LaPadula model, which enforces the system's mandatory integrity rules. Just as the system's mandatory security rules protect information from unauthorized disclosure, the

system's mandatory integrity rules protect information from unauthorized modification.

The system's mandatory integrity policy enables the security administrator or developer to establish highly protected execution domains in which executables may read the files they need while those files remain protected from modification by unauthorized logic or malicious code. The system supports 8 hierarchical role-based integrity classifications and 16 independent non-hierarchical need-to-know integrity categories/compartments.

*Simple Integrity:* Subject may read or execute object (e.g., data file) only when object's integrity level dominates subjects.

*Integrity\*Property:* Subject may write object only when subject's integrity level dominates object's. XTS-400 integrity\*property allows subject to write object only when subject and object integrity levels are the same, preventing lower-integrity subjects from writing higher-integrity objects (which could be considered trustworthy by other software) they cannot later access.

### 5.5 Discretionary Access Controls

The XTS-400 enforces a discretionary access policy whereby access to an object is assigned by the object's owner according to the identity of subjects associated with the object and/or groups to which those subjects belong.

An object's discretionary access information includes up to seven identifiers: for the object's owner, owner group(s), other allowed groups, "world"; and the read, write, execute permissions allowed to those users.

*Access Modes:* Subject may access object in only those mode(s) granted by object's owner. Each object is assigned read, write, execute permissions for object's owner, owner's group(s), members of other groups allowed by owner, and all others ("world" permissions).

The TOE enforces the following series of rules to determine whether a subject should be granted discretionary access to an object.

1. If subject owns object, use specified owner permissions; if not
2. If subject has entry in system's Access Control List, use those permissions; if not

3. If subject's group is same as object's group, use specified group permissions; if not
4. If subject's group exists in ACL, use group ACL permissions; if not
5. Use specified "world" permissions.

## **5.6 Subtype Control**

Processes, devices, and file system objects are controlled by subtype in addition to the mandatory and discretionary controls. Subtypes are non-hierarchical. They can be employed by trusted applications to separate applications (e.g., such as stages in a guard), even if those applications run with the same owner and at the same mandatory level.

This page intentionally left blank

## 6 HARDWARE

Since 1995, a series of XTS-300 Trusted Computer System releases provided customers in the Defense, Intelligence, Diplomatic, and Law Enforcement communities with a National Security Agency-evaluated Class B3 Trusted Server running on Intel server-class hardware. With each hardware migration, the XTS-300 demonstrated increasingly higher performance to match the hardware improvements within the Intel X86 computing architecture. The XTS-400 will follow the same path, beginning with the Intel Pentium® family “Prestonia” Xeon™ 2.8 GHz CPUs.

The XTS-400 is available initially in tower case and rack mounted configurations. Typical later versions which have become available include high density, TEMPEST, and Zone versions. Each is designed with the highest quality components selected specifically for their support of industry standards, reliability, and speed. Careful configuration control and testing are applied to assure delivery of a consistent, reliable, and secure product.

Table 2 provides more detail on the XTS-400 Model 2800 standard hardware configuration. The STOP operating system provides support for the following hardware peripherals:

Table 2. Hardware Details of the XTS-400 Model 2800

CPU type	XEON 2800 MHz or greater
Max number of CPUs	1 initially, designed for 2
Memory	2Gbytes physical (768MB in initial release), 4GB per process virtual
SCSI	High speed SCSI-160
Hard disk transfer speed	160 MB/sec
Hard disk rotation speed	15,000 RPM (except 300GB drive)
Hard disk size	18.2GB standard; upgradeable to 300GB
Video	SVGA video up to 1600 * 1200
Networking	dual-mode 10BaseT/100BaseT Ethernet network, Gigabit in future release
File system	Fast File System
Tape drive	12/24GB DDS-3
Case configurations	Standard tower, 5U rack-mount
Other Peripherals	<ul style="list-style-type: none"> <li>• Standard monitor is 17" SVGA flat screen up to 1600 * 1200</li> <li>• 1U flip up flat panel monitor/keyboard/touchpad optional for rack-mounts. Up to 1024 * 768</li> <li>• LVD monitor</li> <li>• CD-ROMs</li> <li>• PCMCIA PC-Card readers supporting FORTEZZA® encryption devices</li> <li>• floppy drive</li> <li>• network interface controllers</li> <li>• parallel printers</li> <li>• keyboard</li> <li>• mouse or touchpad</li> <li>• serial terminals</li> <li>• UPS including automatic system shutdown software</li> </ul>

This page intentionally left blank

## 7 XTS-300/XTS-400 APPLICATIONS

Trusted applications on the XTS-300™ have significantly aided in the maturation of multilevel security to the point where MLS systems are being deployed widely in operational configurations at low risk, with significant payoff.

Numerous certification and accreditation efforts have been completed of now-operational Trusted Applications— several developed by BAE - IT—that have used the XTS-300™ as their high-assurance platform. These applications include trusted guards developed for the NSA, DISA, the State Department, the Air Force, the FBI, and the Department of Energy. Several more accreditations of XTS-300™ applications are underway.

Most of these applications are Trusted Guards designed to allow the strictly controlled sharing of information among networks operating at different sensitivity levels (e.g., Classifications) and/or “needs-to-know” (categories or compartments). In addition, BAE - IT has developed two “enabling technologies” for building Trusted Guard applications for the XTS-300 and XTS-400: the DataSync Guard and the Standard Automated Guard Environment (SAGE).

With the introduction of STOP 6, the development of Trusted Applications became much easier. Applications can be created entirely on “real” Linux systems, created entirely on an XTS development system, or created partially on both. (Of course, the security related elements of trusted applications can only be tested on an XTS-400 because only the STOP operating system provides the security APIs for use by the application. But the application can be built and tested on “real” Linux less the trusted security relevant piece of code, then that security relevant code can be inserted and the application copied to the XTS for test execution.) In any of these development environments, an entirely new and richer set of rapidly evolving development tools are available to the programmer and designer.

### 7.1 Guard-Enabling Technologies

#### 7.1.1 DataSync Guard

BAE - IT’s customizable DataSync Guard represents a new generation of Trusted Guard application: a TCP/IP socket-based High Assurance Guard. The DataSync Guard strictly enforces the security policies governing the connection-oriented transfer of data

between systems that reside on separate system-high networks at different classification (or sensitivity) levels. The DataSync Guard achieves near-real-time data transfers. Originally conceived to enable the reliable synchronization of databases operating at different sensitivity levels, the DataSync Guard is communications protocol independent, and can handle ASCII, HTML, or well-formed binary data flowing between any two systems that can transfer their data over socket connections to the Guard.

By replacing “store and forward” file transfer protocols with TCP/IP sockets as its data transfer mechanism, and performing most of its processing in memory (eliminating the need to write to or read from disk or resynchronize the file system each time data are passed from one Guard subroutine to another), the DataSync Guard reduces the latency delay of transactions for applications requiring maximum data throughput.

The DataSync Guard can support complex filter profiles to mediate data transfers among databases on up to four different single-level system-high networks. The DataSync Guard can filter ASCII and/or well-formed binary data, checking both the correct formatting of the header of the database transaction, and performing security checks on the content of each data element. The DataSync Guard can implement multiple “if-then-else” actions and sophisticated dirty word/clean word searches.

#### 7.1.2 Standard Automated Guard Environment

The Standard Automated Guard Environment (SAGE™) is a set of design concepts, interface definitions, executable code and accreditation documentation. SAGE is a development environment for building connectionless (store-and-forward) Trusted Guards.

SAGE minimizes the coding required to implement Trusted Guards by providing the common elements (processes, libraries, etc.) that most Guards require. SAGE eases certification and accreditation of these Guards by absolutely minimizing the Guard’s TOE. SAGE provides a well-structured framework within which programmers can build Trusted Guard applications more quickly and easily than developing those applications “from scratch”. The general objective of a SAGE Guard is to securely, automatically, and efficiently allow a restricted flow of data between two systems or networks with different security characteristics. While the security

policy can be customized by the Guard developer, SAGE has been designed to accurately enforce that policy and to protect data from unauthorized disclosure or modification while the data resides on the XTS-400 system.

SAGE has been developed in standard ANSI C, and documented using trusted software principles to ease the burden of accreditation. Several SAGE guards have already been accredited, including the Defense Information Infrastructure Guard, the State Department Unclassified Telegram Guard Processor, the Air Force F-22 Secure Interface System, and the Military Sealift Command Database Replication Guard and future accreditation efforts may benefit from the previous work.

For more information on SAGE, consult the *Standard Automated Guard Environment Technical Overview*.

## 8 WHY WOULD YOU USE THE XTS-400?

Why use XTS-400 instead of other products for a security demanding application?

- Because it promises a more robust set of security enforcement features and a higher assurance that those features do what they are documented to do.
- Because STOP 6.0 is built on an existing, proven foundation.
- Because it is built by a group which has successfully taken multiple products through multiple successful evaluations.
- Because multiple successful accreditations have been performed on applications hosted on the XTS. BAE - IT is experienced in assisting the application to achieve accreditation.
- Because BAE - IT has applications development experience that can provide everything from product to a turn key application. Services familiar with building security enforcing applications are available from BAE - IT to help you with everything from requirements analysis to software development to accreditation support to deployment support to education to life cycle maintenance. BAE Systems can supply you with as little or as much as you want.
- Because the basic design of the XTS has been repeatedly analyzed and penetration tested by security experts.
- Because the people who wrote the operating system are committed to the support and enhancement of the product. The help line is answered by a developer.
- Because custom devices or functionality are possible within the security architecture. The group has a history of successfully developing “special” capabilities in response to customer requirements.
- Because software updates and “fixes” are produced using the same processes (tools, CM processes, documentation changes, designs, peer reviews, etc) that produced the original product.
- Because the hardware and software has a history of successful updates and migration paths that are customer friendly.
- Because BAE - IT can support the development, deployment, hardware, software and application for the life cycle of the program’s needs including needed improvements.
- Because the secure product business is one of BAE - IT’s core strengths and you can be assured of life cycle support for your program.

© Copyright 2002-2024 by BAE System Information Technology, LLC. XTS-300, XTS-400, SAGE and STOP are trademarks of BAE System Information Technology, LLC. UNIX is a trademark of AT&T UNIX Systems Laboratories. FORTEZZA is a registered trademark of the U.S. National Security Agency. Linux is a trademark of Linus Torvalds. Red Hat and RPM are trademarks of Red Hat Software Inc. Intel, Pentium are registered trademarks and Xeon is a trademark of the Intel Corp.

**BAE SYSTEMS**