Cray UNICOS/mp Operating System Version 2.4.15 on Cray X1 hardware Security Target

Version 1.0

August 31, 2004

Prepared for:

Cray Inc.

1340 Mendota Heights Road

Mendota Heights, MN 55120

Prepared By:

Science Applications International Corporation

Common Criteria Testing Laboratory

7125 Columbia Gateway Drive, Suite 300
Columbia, MD 21046

The UNICOS/mp operating system is derived from UNIX System V. This operating system is also based in part on the Fourth Berkeley Distribution (BSD) under license from the The Regents of University of California.

LIST OF TABLES

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is provided by Cray Inc. The TOE is the Cray UNICOS/mp operating system on Cray X1 hardware.

The Security Target contains the following additional sections:

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

1.1 Security Target, TOE and CC Identification

ST Title – Cray UNICOS/mp Operating System Version 2.4.15 on Cray X1 hardware Security Target

ST Version – Version 1.0

ST Date – August 31, 2004

TOE Identification – Cray UNICOS/mp Operating System version 2.4.15 on Cray X1 hardware

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 2.1, August 1999, ISO/IEC 15408.

1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional requirements, Version 2.1, August 1999, ISO/IEC 15408-2.
    - Part 2 Conformant
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance requirements, Version 2.1, August 1999, ISO/IEC 15408-3.
    - Part 3 Conformant
    - EAL 2 augmented with ALC_FLR.1
- Strength of Function Claim
    - The minimum strength of function level for the security functional requirements is SOF-medium.

1.3 Conventions, Terminology, Acronyms

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

1.3.1 Conventions

The following conventions have been applied in this document:

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

1.3.2 Acronyms

CC Common Criteria

CNS Cray Network System

CPES Cray Programming Environment Server

CWS Cray Workstation

EAL Evaluation Assurance Level

IT Information Technology

OS Operating System

SFP Security Function Policy

SFR Security Function Requirement

SPC System Port Channel

ST Security Target

TOE Target of Evaluation

TSC TSF Scope of Control

TSF TOE Security Functions

TSP TOE Security Policy

2. TOE Description

The TOE includes the UNICOS/mp operating system, Cray X1 hardware, and those applications necessary to manage, support and configure the operating system.

2.1 Product Type

The TOE is a supercomputer, providing a scalable platform for heavy calculating applications not suited to more traditional computers and servers. Cray X1 systems utilize vector processors, shared memory, and a modernized vector instruction set. A Cray X1 system has high memory bandwidth and scalable system software.

The UNICOS/mp operating system provides a UNIX-based interface and command structure for the administration and use of the Cray X1.

2.2 Product Description

The TOE is comprised of several components, both hardware and software. The hardware is grouped into three sections: the Cray X1 mainframe, the Cray X1 I/O architecture, and the Cray X1 support system. The software is composed of the UNICOS/mp operating system that runs on the Cray X1 mainframe. The following components that run in the I/O architecture, the Cray Programming Environment Server (CPES), Cray Network Subsystem (CNS), and support system sections (Cray Workstation, CWS), are not included within the scope of the TOE.

The primary hardware component is the Cray X1 mainframe itself, which is composed of nodes. A node is a group of 4 processors and shared memory, along with 4 high-bandwidth interconnection modules (SPC I/O Ports). Multiple nodes can be interconnected with a minimum of 2 nodes required to create a useable Target of Evaluation (TOE). Interconnected multiple nodes provide shared processing capacity and memory across all nodes.

The Cray X1 I/O architecture includes the following components, which are housed in multiple cabinets, though only the RAID subsystem is included within the TOE:

- I/O drawers that convert the SPC protocol used by the nodes to Fibre Channel protocol used by various peripheral devices. Each drawer supports four SPC channels to the mainframe and up to 16 Fibre Channels to peripheral devices.
- Cray Programming Environment Server (CPES) that runs the tools used by the programmer for program development.
- Cray Network Subsystem (CNS) that connects a Cray X1 system to the customer's networks; supported protocols include Gigabit Ethernet and HIPPI.
- RAID subsystem that provides disk storage for the Cray X1 system.

The CPES provides the Cray X1 with the ability to offload compilation of code from the mainframe, allowing for a distribution of programming environment processes away from the resources of the mainframe. The CPES is based on a dedicated Sun Solaris server.

The CNS provides network connectivity for the Cray X1 mainframe, packaging network packets into higher bandwidth chunks that are more efficient for the Cray X1 to handle, and reducing latency issues for the mainframe. The CNS is built on a Dell server running Linux.

The Cray X1 support system consists primarily of the Cray Workstation, which is used to administer the mainframe. The CWS is built on a Sun Solaris server. The CPES and CNS are physically housed in one rack, possibly along with any attached RAID arrays.

The UNICOS/mp operating system that runs on the mainframe is based upon IRIX 6.5, optimized for use on Cray X1 hardware. The UNICOS/mp operating system provides for separate hardware modes of operation, having kernel and user modes.

2.3 Product Features

The TOE provides the following capabilities:

TRUE SINGLE SYSTEM IMAGE (SSI)

To application developers, users, and administrators, a Cray X1 system is a single system, regardless of its size. Applications can read and write memory and files regardless of the processor on which the application is running. Users have a single login, and administrators have a single point of management and control for the entire system.

SCHEDULING ALGORITHMS

The UNICOS/mp operating system uses sophisticated scheduling algorithms to effectively schedule parallel applications, user commands, and operating system processes to share a Cray X1 system.

ACCELERATED APPLICATION MODE AND MIGRATION

Parallel applications can be run in accelerated application mode, which makes use of system memory mapping hardware to create logically contiguous nodes that help optimize application performance. To ensure maximum availability of contiguous nodes, the UNICOS/mp operating system transparently migrates applications within the system.

FLEXIBLE SYSTEM PARTITIONING

Some customers may find it advantageous to partition their Cray X1 system. Partitioning allows system administrators to divide a system into two or more separate systems, each with an independent operating system image. These partitions can be separately booted, dumped, halted, and so on, without impacting other running partitions. Halted partitions can be combined into larger, or split into smaller partitions.

2.4 Security Environment TOE Boundary

The TOE includes the following physical and logical boundaries.

2.4.1 Physical Boundaries

The Cray X1 mainframe running the UNICOS/mp operating system can be decomposed into components based upon the system architecture. Among these components, some are considered to be a part of the IT environment. The following components are considered parts of the IT environment: the CPES, the CWS and the CNS. The CWS provides the administrator with access to the management functions that exist on the Cray X1 mainframe. Users access the TOE via terminals or workstations over a network connection.

Within the TOE boundary is the core Cray X1 mainframe (all nodes and RAID disk arrays) and the UNICOS/mp operating system. The TOE can then be decomposed into three subsystems: the hardware, the UNICOS/mp kernel, and the UNICOS/mp user environment.

The UNICOS/mp operating system provides a protected kernel mode and an untrusted user mode. The kernel mode of the system provides core functions of the system, including I/O to the hardware and system security functions. The kernel provides administrative tools used in the management of the included security functions, Identification and Authentication and Access Control policies. The user mode provides the command structure available to users of the system, within the bounds set by the administrators (e.g. within the security bounds placed on user access rights).

2.4.2 Logical Boundaries

The TOE security functions are derived from the Controlled Access Protection Profile (CAPP), though CAPP compliance is not claimed.

2.4.2.1 User data protection

The UNICOS/mp operating system enforces a discretionary access control (DAC) policy on all subjects and objects. Discretionary Access Control (DAC) is the access control mechanism by which a person who created an object can

choose to grant a specific type of access to another user or group of users. Access to objects is controlled based solely on the identity of the user and the identity of the object. The implementation of a DAC policy is accomplished by the association of attributes that are specific to the type of object (also called resource). Access to the resource then initiates a check of the attributes to determine whether and what type of access is granted to the resource.

2.4.2.2 Identification and authentication

Identification and Authentication security functions provided by the UNICOS/mp operating system relate primarily to user attributes and authentication. Through the provided security management functions (see next section), an administrator is able to create a user and associate various attributes to that account. These attributes can include items such as authentication data, and group memberships. Based upon the associated attributes, a user can then authenticate to the UNICOS/mp operating system, and access resources for which access has been granted.

The UNICOS/mp operating system also provides system security by not allowing any actions to be taken on the system without a successful user authentication, providing that only authorized users have access to the resources of the TOE. In addition, all actions are bound to a user, preventing users from interfering or accessing each others work and/or resources.

2.4.2.3 Security management

The UNICOS/mp operating system provides a collection of tools for authorized administrators to manage user accounts and data. These include tools for the management of authentication data, as well as user tools for the self-management of passwords. The UNICOS/mp operating system also provides tools for the creation and management of DAC policies, allowing authorized to manage ACLs and permission bits of files, directories and other resources. The UNICOS/mp operating system provides the role of root for the authorized administrator.

2.4.2.4 Protection of the TSF

The UNICOS/mp operating system and associated hardware protects itself by assuring that the security policies are always enforced, providing a basis for all other security functionality, including domain separation, which prevents untrusted processes from interfering with the TOE. This ensures that kernel mode processes cannot be tampered with by untrusted processes (such as user processes).

3. Security Environment

This section defines the security policies the TOE, in conjunction with its environment, is intended to fulfill as well as usage assumptions about the TOE's intended environment.

3.1 Organizational Policies

P.AUTHORIZED_USERS Only those users who have been authorized to access the information within the system may access the system.

P.NEED_TO_KNOW The system must limit the access to, modification of, and destruction of the information in protected resources to those authorized users who have a 'need to know' for that information.

3.2 Assumptions

A.LOCATE	The processing resources of the TOE will be located within controlled access facilities, which will prevent unauthorized physical access.
A.MANAGE	There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.
A.NO_EVIL_ADM	The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the administrator documentation.
A.PEER	Any other systems with which the TOE communicates are assumed to be under the same management control and operate under the same security policy constraints.
A.PROTECT	The TOE hardware and software critical to security policy enforcement will be protected from unauthorized physical modification.

4. Security Objectives

This section defines the security objectives of the TOE and its supporting environment. Security objectives, categorized as either IT security objectives or non-IT security objectives, reflect the stated intent to counter identified threats and/or comply with any organizational security policies identified. All of the identified threats and organizational policies are addressed under one of the categories below.

4.1 Security Objectives for the TOE

O.AUTHORIZATION The TSF must ensure that only authorized users gain access to the TOE and its resources.

O.DISCRETIONARY_ACCESS The TSF must control access to resources based on identity of users. The TSF must allow authorized users to specify which users may access which resources.

O.ENFORCEMENT The TSF must be designed and implemented in a manner that ensures the organizational policies are enforced in the target environment.

O.MANAGE The TSF must provide all the functions and facilities necessary to support the authorized administrators who are responsible for the management of TOE security.

4.2 Security Objectives for the Environment

O.INSTALL Those responsible for the TOE must ensure that the TOE is delivered, installed, managed, and operated in a manner that maintains IT security objectives.

O.PHYSICAL Those responsible for the TOE must ensure that those parts of the TOE critical to security policy are protected from physical attack that might compromise IT security objectives.

O.CREDEN Those responsible for the TOE must ensure that all access credentials, such as passwords or other authentication information, are protected by the users in a manner which maintains IT security objectives.

O.STAFF Those responsible for the TOE will be competent to manage it and the security of the information it contains.

O.PEER Any other systems with which the TOE communicates must be under the same management control and operate under the same security policy constraints.

5. IT Security Requirements

5.1 TOE Security Functional Requirements

The following table describes the SFRs that are candidates to be satisfied by the UNICOS/mp operating system.

Requirement Class	Requirement Component
FDP: User data protection	FDP_ACC.1: Subset access control
	FDP_ACF.1: Security attribute based access control
FIA: Identification and authentication	FIA_ATD.1: User attribute definition
	FIA_UAU.2: User authentication before any action
	FIA_UID.2: User identification before any action
	FIA_USB.1: User-subject binding
FMT: Security management	FMT_MSA.1: Management of security attributes
	FMT_MSA.3: Static attribute initialization
	FMT_MTD.1a: Management of TSF data
	FMT_MTD.1b: Management of TSF data
	FMT_MTD.1c: Management of TSF data
	FMT_SMF.1: Specification of Management Functions *(per International Interpretation #65)*
	FMT_SMR.1: Security roles
FPT: Protection of the TSF	FPT_RVM.1: Non-bypassability of the TSP
	FPT_SEP.1: TSF domain separation

Table 1: Security Functional Components

5.1.1 User data protection (FDP)

5.1.1.1 Subset access control (FDP_ACC.1)

FDP_ACC.1.1 The TSF shall enforce the [**Discretionary Access Control Policy**] on [**all operations among processes acting on behalf of users and the following objects: files, directories, named pipes, and processes**].

5.1.1.2 Security attribute based access control (FDP_ACF.1)

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

1) **the user identity, and group membership(s) associated with a process and**

2) **the following access control attributes associated with an object: UNIX Permission bits, ACL, and Object Ownership**]. *(per International Interpretation #103)*

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

1) **If the object has an explicit ACL, access is granted if one of the following is true:**

   a) **An ACL entry explicitly grants access to a user,**

b) An entry explicitly grants access to a group of which the subject is a member and the access has not been denied by a previous entry in the ACL, or

c) An ACL entry explicitly grants access to the world and the access has not been denied by a previous entry in the ACL.

2) If an object has permission bits, access is granted if one of the following is true:

a) If the subject is the owner of the object and the object's owner UNIX permission bits indicate that the operation required accesses are allowed.

b) If the subject is a member of the object owning group and the object's group UNIX permission bits indicate that the operation required accesses are allowed, or

c) If the object's world UNIX permission bits indicate that the operation required accesses are allowed.

3) The process's effective or real UID or the process's group ID is the same as the object owner and the operation is performed on a process].

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [**If a process has the UID of 0 (i.e., 'root'), the TSF shall authorize access of the process to any object, even if such access is disallowed by FDP_ACF.1.2**].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the [**no additional rules**].

5.1.2 Identification and authentication (FIA)

5.1.2.1 User attribute definition (FIA_ATD.1)

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: [**User identity, group identities, authentication data, and security-relevant roles (i.e., 'root')**].

5.1.2.2 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.

5.1.2.3 User identification before any action (FIA_UID.2)

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

5.1.2.4 User-subject binding (FIA_USB.1)

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

5.1.3 Security management (FMT)

5.1.3.1 Management of security attributes (FMT_MSA.1)

FMT_MSA.1.1 The TSF shall enforce the [**Discretionary Access Control Policy**] to restrict the ability to [***modify***] the security attributes [**access control attributes associated with protected objects (i.e., identified in FDP_ACC.1)**] to [**the object owner, or 'root'**].

5.1.3.2 Static attribute initialization (FMT_MSA.3)

FMT_MSA.3.1 The TSF shall enforce the [**Discretionary Access Control Policy**] to provide [***restrictive***] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [**creator or authorized administrator**] to specify alternative initial values to override the default values when an object or information is created.

5.1.3.3 Management of TSF data (FMT_MTD.1a)

FMT_MTD.1a.1 The TSF shall restrict the ability to [***initialize and modify***] the [**user security attributes, other than authentication data**] to [**authorized administrators**].

5.1.3.4 Management of TSF data (FMT_MTD.1b)

FMT_MTD.1b.1 The TSF shall restrict the ability to [***modify***] the [**user authentication data**] to [**authorized administrators and the user associated with the authentication data**].

5.1.3.5 Management of TSF data (FMT_MTD.1c)

FMT_MTD.1c.1 The TSF shall restrict the ability to [***initialize***] the [**user authentication data**] to [**authorized administrators**].

5.1.3.6 Specification of Management Functions *(per International Interpretation #65)* (FMT_SMF.1)

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: [**manage user security attributes**]. *(per International Interpretation #65)*

5.1.3.7 Security roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles [**authorized administrator, users authorized by the Discretionary Access Control Policy to modify object security attributes, and user authorized to modify their own authentication data**].

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

5.1.4 Protection of the TSF (FPT)

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

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

5.1.4.2 TSF 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.

5.2 Security Functional Requirements for the IT Environment

This Security Target places no security functional requirements on the IT environment.

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL 2 augmented with ALC_FLR.1 components as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components.

Requirement Class	Requirement Component
ACM: Configuration management	ACM_CAP.2: Configuration items
ADO: Delivery and operation	ADO_DEL.1: Delivery procedures
	ADO_IGS.1: Installation, generation, and start-up procedures
ADV: Development	ADV_FSP.1: Informal functional specification
	ADV_HLD.1: Descriptive high-level design
	ADV_RCR.1: Informal correspondence demonstration
AGD: Guidance documents	AGD_ADM.1: Administrator guidance
	AGD_USR.1: User guidance
ALC: Life cycle support	ALC_FLR.1: Basic flaw remediation
ATE: Tests	ATE_COV.1: Evidence of coverage
	ATE_FUN.1: Functional testing
	ATE_IND.2: Independent testing - sample
AVA: Vulnerability assessment	AVA_SOF.1: Strength of TOE security function evaluation
	AVA_VLA.1: Developer vulnerability analysis

Table 2: EAL 2 augmented with ALC_FLR.1 Assurance Components

5.3.1 Configuration management (ACM)

5.3.1.1 Configuration items (ACM_CAP.2)

ACM_CAP.2.1d The developer shall provide a reference for the TOE.

ACM_CAP.2.2d The developer shall use a CM system.

ACM_CAP.2.3d The developer shall provide CM documentation.

ACM_CAP.2.1c The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.2.2c The TOE shall be labelled with its reference.

ACM_CAP.2.3c The CM documentation shall include a configuration list.

ACM_CAP.2.RI-3 The configuration list shall uniquely identify all configuration items that comprise the TOE. *(per International Interpretation #3)*

ACM_CAP.2.4c The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.2.5c The CM documentation shall describe the method used to uniquely identify the configuration items.

ACM_CAP.2.6c The CM system shall uniquely identify all configuration items.

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

5.3.2 Delivery and operation (ADO)

5.3.2.1 Delivery procedures (ADO_DEL.1)

ADO_DEL.1.1d The developer shall document procedures for delivery of the TOE or parts of it to the user.

ADO_DEL.1.2d The developer shall use the delivery procedures.

ADO_DEL.1.1c The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.

ADO_DEL.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

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

ADO_IGS.1.1d The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.

ADO_IGS.1.1c The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation and start-up of the TOE. *(per International Interpretation #51 (rev 1))*

ADO_IGS.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADO_IGS.1.2e The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

5.3.3 Development (ADV)

5.3.3.1 Informal functional specification (ADV_FSP.1)

ADV_FSP.1.1d The developer shall provide a functional specification.

ADV_FSP.1.1c The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV_FSP.1.2c The functional specification shall be internally consistent.

ADV_FSP.1.3c The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.

ADV_FSP.1.4c The functional specification shall completely represent the TSF.

ADV_FSP.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.1.2e The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.2 Descriptive high-level design (ADV_HLD.1)

ADV_HLD.1.1d The developer shall provide the high-level design of the TSF.

ADV_HLD.1.1c The presentation of the high-level design shall be informal.

ADV_HLD.1.2c The high-level design shall be internally consistent.

ADV_HLD.1.3c The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.1.4c The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV_HLD.1.5c The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV_HLD.1.6c The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV_HLD.1.7c The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV_HLD.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_HLD.1.2e The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.3 Informal correspondence demonstration (ADV_RCR.1)

ADV_RCR.1.1d The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

ADV_RCR.1.1c For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

ADV_RCR.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4 Guidance documents (AGD)

5.3.4.1 Administrator guidance (AGD_ADM.1)

AGD_ADM.1.1d The developer shall provide administrator guidance addressed to system administrative personnel.

AGD_ADM.1.1c The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

AGD_ADM.1.2c The administrator guidance shall describe how to administer the TOE in a secure manner.

AGD_ADM.1.3c The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD_ADM.1.4c The administrator guidance shall describe all assumptions regarding user behaviour that are relevant to secure operation of the TOE.

AGD_ADM.1.5c The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD_ADM.1.6c The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_ADM.1.7c The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD_ADM.1.8c The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

AGD_ADM.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4.2 User guidance (AGD_USR.1)

AGD_USR.1.1d The developer shall provide user guidance.

AGD_USR.1.1c The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD_USR.1.2c The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD_USR.1.3c The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD_USR.1.4c The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behaviour found in the statement of TOE security environment.

AGD_USR.1.5c The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD_USR.1.6c The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

AGD_USR.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.5 Life cycle support (ALC)

5.3.5.1 Basic flaw remediation (ALC_FLR.1)

ALC_FLR.1.1d The developer shall provide flaw remediation procedures addressed to TOE developers.

ALC_FLR.1.1c The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.1.2c The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.1.3c The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.1.4c The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.6 Tests (ATE)

5.3.6.1 Evidence of coverage (ATE_COV.1)

ATE_COV.1.1d The developer shall provide evidence of the test coverage.

ATE_COV.1.1c The evidence of the test coverage shall show the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

ATE_COV.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.6.2 Functional testing (ATE_FUN.1)

ATE_FUN.1.1d The developer shall test the TSF and document the results.
ATE_FUN.1.2d The developer shall provide test documentation.

ATE_FUN.1.1c The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE_FUN.1.2c The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE_FUN.1.3c The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.4c The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.5c The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

ATE_FUN.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.6.3 Independent testing - sample (ATE_IND.2)

ATE_IND.2.1d The developer shall provide the TOE for testing.

ATE_IND.2.1c The TOE shall be suitable for testing.

ATE_IND.2.2c The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

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

ATE_IND.2.2e The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified.

ATE_IND.2.3e The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

5.3.7 Vulnerability assessment (AVA)

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

AVA_SOF.1.1d The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.

AVA_SOF.1.1c For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.

AVA_SOF.1.2c For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.

AVA_SOF.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_SOF.1.2e The evaluator shall confirm that the strength claims are correct.

5.3.7.2 Developer vulnerability analysis (AVA_VLA.1)

AVA_VLA.1.1d The developer shall perform a vulnerability analysis. *(per International Interpretation #51 (rev 1))*

AVA_VLA.1.2d The developer shall provide vulnerability analysis documentation. *(per International Interpretation #51 (rev 1))*

AVA_VLA.1.1c The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for obvious ways in which a user can violate the TSP. *(per International Interpretation #51 (rev 1))*

AVA_VLA.1.2c The vulnerability analysis documentation shall describe the disposition of obvious vulnerabilities. *(per International Interpretation #51 (rev 1))*

AVA_VLA.1.3c The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE. *(per International Interpretation #51 (rev 1))*

AVA_VLA.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VLA.1.2e The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure obvious vulnerabilities have been addressed.

6. TOE Summary Specification

This section describes the security functions and associated assurance measures.

6.1 TOE Security Functions

6.1.1 User data protection

The UNICOS/mp operating system enforces a discretionary access control (DAC) policy on all subjects and objects. Discretionary Access Control (DAC) is the mechanism by which an access to data is controlled based solely on the identity of the user and the resource. The implementation of DAC is accomplished by association of attributes, which are specific to the type of resource. This section first identifies the DAC attributes of subjects and each object. The DAC policies are specific to the type of object and are described following the attribute identification.

6.1.1.1 DAC Attributes

Following are the DAC-related attributes of a subject:

- Real UID
- Saved UID
- Saved GID
- Process group ID and session ID
- Process group leader ID
- Effective UID
- Real GID
- Effective GID
- Group list
- Process umask

Following are the DAC-related attributes of a file system object:

- Owner
- Owning Group
- Protection Mode
- Access Control List (optional)
- Default Access Control List (optional, relevant only for directories)

Following are the DAC-related attributes of a process object:

- Real User ID
- Effective User ID
- Saved User ID
- Process ID
- Process Group ID
- Process Group Structure
- Session Structure

6.1.1.2 DAC Permissions Checking for File System Objects

Permission checking relies on several object attributes. Two of those attributes are permission bits (i.e., protection mode) and Access Control Lists (ACLs). This section describes permission bits and ACLs and their access checking algorithms.

Permission bits divide accesses into three categories and users into three relative groups. The three categories of permissions are read, write, and execute. They are denoted as "r" for read, "w" for write, and "x" for execute. The three relative groups are the owner of the file, the owner's group, and every other user.

Permissions checking on permission bits checks for user permission first, followed by group, followed by other users. Permission is granted by the first permission set matched.

The UNICOS/mp operating system includes an additional DAC facility, Access Control Lists (ACLs) for files and directories. The ACL is an optional component of the DAC. ACLs are used to provide more fine-grained protection than the group permissions.

Any system file or directory may have an ACL that governs its discretionary access. This ACL is referred to as the access ACL for the file or directory. In addition, a directory may have an associated ACL that is applied to any file or subdirectory created within that directory. This ACL is referred to as a default ACL. The default ACL on a directory is not used to determine discretionary access to the directory. When a file or subdirectory is created within that directory, the file or subdirectory is assigned a new access ACL according to the following rules:

1) Starts with a copy of the parent directory's default ACL as its access ACL.
2) The the user::, group::, other::, and mask:: entries are taken from thefile's permission bits, with the process' umask applied.
3) If the parent directory's default ACL contains a mask entry, the access ACL mask entry is set from the file's group permission bits.
4) If the parent directory's default ACL does not contain a mask entry, then the new file's ACL group entry is set from the file's group permission bits.

6.1.1.3 DAC Permissions Checking for Process Objects

A process can always read and write its own data in user space. A process can also always read its own attributes. It can change its real UID and effective UID if the requested value is equal to its real or saved UID. A process may change its effective GID if the GID is equal to the real or saved GID.

There is one instance where processes access one another. One process can always read the attributes of another process if it can name the target process using its PID.

6.1.1.4 DAC Initialization and Revocation

For files and directories, if an ACL is used, permissions are assigned as described in Section 6.1.1.2. If ACLs are not used, the creating process umask is used to assign initial permission. A umask contains the initial permission settings and may be set as restrictively as the subject wants. The various IDs associated with objects are taken from the creating subject’s attributes. Process attributes are assigned at logon or are taken from a parent process. Object owners or administrators can modify the security attributes of objects.

If a subject changes the permissions of an object, the changes take affect on the next access check against the object. So, if a subject has a file descriptor open for an object and attempts to use the file descriptor, the old permissions will remain in affect until the subject closes the object. If the subject closes the object and then attempts to re-open it, the new permissions would then be enforced.

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

- FDP_ACC.1
- FDP_ACF.1
- FMT_SMR.1
- FMT_MSA.1

6.1.2 Identification and authentication

When workstations are connected to the TOE, a user may request to authenticate (connecting through telnet) to the TOE to initiate processes on the TOE. This process prompts the user for an initial user name and then passes the name to the login command. The login command then prompts for a password. The login command will then continue to prompt for username / password pairs until a correct pair is entered or a configurable maximum number of attempts is exceeded. At this point, a configurable time delay occurs, and the login command exits. The user will not be allowed to perform any actions on the TOE until a successful authentication.

The /etc directory maintains the user attribute information. The identification and authentication subsystem maintains a file within the /etc directory that provides basic authentication data such as user name, user ID, and password. The user name may be an untrusted user or a role (i.e. non-administrative).

This information is supplemented by the following related database: group membership (/etc/group)

The user is never allowed to see the plain text form of a password, or alter the password file directly. Only an administrator can manage user attributes (other than a user's password). When an administrator makes a change to a user attribute using the passmgmt, the change takes affect on the next login. To ensure an immediate change, the administrator must require the user to logoff.

Once a user has been authenticated, process groups are used to associate users with subjects on the operating system. A subject is an entity within the TSC that causes operations to be performed. A process group is a set of processes that potentially all share the same address space. The degenerate case of a process group is a single process. A process is one thread of execution within a process group. Each process in a process group has its own stack. Every process in a process group potentially has access to the data (including the stack) of all the others. Other process attributes that may be shared include the address space, open file table, the current and root directories, the umask, the maximum file size, and the real and effective user identifiers (UIDs) and group identifiers (GIDs).

An existing subject can create a new subject by using the fork or exec system calls. A fork creates a new process that is a copy of the creating process. The new process is a child of the creator and has a new unique process ID. When a process within a process group performs an exec, it becomes the first process in a new process group and therefore, a new subject. A sproc system call is used to create a new process within an existing process group, but does not create a new subject unless and until the new process performs an exec system call, which changes its security attributes. All attributes are assigned to a subject at creation time.

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

- FIA_ATD.1
- FIA_UAU.2
- FIA_UID.2
- FIA_USB.1

6.1.3 Security management

The UNICOS/mp operating system supports the administrative role of root, the traditional UNIX administrative user. The root user is permitted to perform all administrative actions including the ability to manage user authentication data and modify the default discretionary access policy. The terms UID=0, super user, root user and authorized administrator are used synonymously with root.

Users are allowed to change their own passwords using the passwd command. Passwords must be constructed to meet the following requirements:

- Each password must have at least six characters.
- Each password must contain at least two alphabetic characters and at least one numeric or special character. In this case, ``alphabetic'' means upper and lower case letters.
- Each password must differ from the user's login name and any reverse or circular shift of that login name. For comparison purposes, an upper case letter and its corresponding lower case letter are equivalent.

- New passwords must differ from the old by at least three characters. For comparison purposes, an upper case letter and its corresponding lower case letter are equivalent.

Administrators assign and can change any user's password using the same passwd command. User changes to passwords using the passwd command are constrained by password aging. Password aging means after a password change, the user may not change the password again for a site configured minimum period of time. This prevents the user from changing away from and back to an expired password too quickly. Once a password has been validated by the passwd command, it is encrypted and written into the password file. The clear text is never written into any file.

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

- FMT_MSA.1
- FMT_MSA.3
- FMT_MTD.1a
- FMT_MTD.1b
- FMT_MTD.1c
- FMT_SMF.1
- FMT_SMR.1

6.1.4 Protection of the TSF

The UNICOS/mp operating system executes instructions in two broad domains. Commands and applications execute in user-mode. In this mode, the memory management system and hardware restrictions on instruction execution isolate processes from each other, from operating system control data, and from direct hardware access. This establishes a strong separation of the instruction streams and data contained in one process from those found in another process.

A thread within a share-group may initiate an operating system request from user-mode through the system call trap mechanism. A system call trap is a software interrupt operation that causes a context switch from user-mode into kernel-mode. In kernel-mode, the thread can only execute the kernel defined code sequence that follows from a system call. This kernel code sequence, however, has unrestricted direct access to kernel control data and the hardware. The kernel-mode and user-mode distinctions are enforced by the hardware.

A process in the UNICOS/mp operating system may have any of three levels of trust at any given time, depending on the code it executes. These three levels of trust define three trust domains for user-mode in the UNICOS/mp operating system system:

- The user domain
- The administrative domain
- The system domain

The user domain contains processes without special authorization, operating on behalf of a single named user. Processes in the user domain may be simple or complex, but they cannot violate DAC restrictions, so they have no particular security function.

The administrative domain contains processes running with or without special authorization operating on behalf of administrators. Processes in the administrative domain always have the potential to assert administrative authority over policy restrictions. While these processes have a security function, that function is entirely under administrative control and, therefore, largely beyond comprehensive analysis. This limits analysis to correctness of function and potential administrative action.

The system domain contains processes executing with special authorization on behalf of potentially non-administrative users. This includes system daemons that provide services like authentication, batch processing, and so forth. It also includes applications that allow a user limited access to protected system resources, a password changing command, for example. Execution in the system domain implies policy enforcement by user-mode

software. Policy enforcement requires well-defined and repeatable behavior, so analysis of system domain software includes a description of the controls placed by the software. Process isolation and memory protection features ensures that these processes have a protected execution environment.

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

- FPT_RVM.1
- FPT_SEP.1

6.2 TOE Security Assurance Measures

6.2.1 Configuration management

The configuration management measures applied by Cray ensure that configuration items are uniquely identified, and that documented procedures are used to control and track changes that are made to the TOE. Cray ensures changes to the implementation representation are controlled and that TOE associated configuration item modifications are properly controlled. Cray performs configuration management on the TOE implementation representation, design, tests, user and administrator guidance, and the CM documentation.

These activities are documented in:

- UNICOS/mp™ 2.4 Release Overview
- Supplement Guide for Configuration Management
- UNICOS/mp 2.4.15 fix package

The Configuration management assurance measure satisfies the following EAL 2 augmented with ALC_FLR.1 assurance requirements:

- ACM_CAP.2

6.2.2 Delivery and Guidance

Cray provides delivery documentation that explains how the TOE is delivered and procedures to identify the TOE, allow detection of unauthorized modifications of the TOE. Cray's delivery procedures state the TOE comes pre-installed by a Cray representative.

These activities are documented in:

- Supplement Guide for Delivery, April 28, 2004

The Guidance Documents provided by Cray include both administrator and user manuals. The administrator manual describes the administrative functions and interfaces, provides guidance on how to administer the TOE securely, and contains warnings about functions and privileges that should be controlled. The user manual describes the functions and interfaces available to non-administrative users, describes the user-accessible security functions, and contains warnings to users about functions that should be controlled.

These activities are documented in:

- NIAP Security Evaluation: UNICOS/mp System Administration, Version 1.0, May 2004 (S-9916-10)
- Supplement Guide for Guidance Documents, June 22, 2004
- UNICOS/mp Administrative Security Overview NIAP Errata 1, July 28, 2004
- UNICOS/mp Installation Guide, Version 2.4, March 2004
- UNICOS/mp Networking Facilities Administration, Version 2.4, March 2004
- UNICOS/mp Disks and File Systems Administration, Version 2.4, March 2004

- UNICOS/mp 2.4 Release Overview, Version 2.4, 2004
- Recommendations for User-Level Security on a Cray X1 UNICOS/mp System NIAP Errata 2, June 1, 2004
- Cray Technology, Inc. XML Web Services Management and XMS Firewall Security Solution, XML Message Server (XMS), Version 3.0

The Delivery and Guidance documents assurance measure satisfies the following EAL 2 augmented with ALC_FLR.1 assurance requirements:

- ADO_DEL.1
- ADO_IGS.1
- AGD_ADM.1
- AGD_USR.1

6.2.3 Development

The Design Documentation provided for the TOE is provided in two documents:

- Man Pages (MAN1, MAN2, MAN3, MAN5, MAN7, MAN8)
  - Man Page Collection: UNICOS/mp™ User Commands
  - Man Page Collection: UNICOS/mp™ System Calls
  - Man Page Collection: UNICOS/mp™ Administrator Commands)
- Architecture Design Documents
- Correspondence Spreadsheet

These documents serve to describe the security functions of the TOE, its interfaces both external and between subsystems, the architecture of the TOE (in terms of subsystems), and correspondence between the available design abstractions (including the ST).

The Development assurance measure satisfies the following EAL 2 augmented with ALC_FLR.1 assurance requirements:

- ADV_FSP.1
- ADV_HLD.1
- ADV_RCR.1

6.2.4 Life cycle support

Cray applies procedures to accept and act upon reported security flaws and requests to correct security flaws. Cray designates specific points of contact for user reports and security related inquiries. The procedures are documented and describe how security flaws are tracked, that for each security flaw a description and status of the correction of the security flaw is provided, that corrective actions are identified for each security flaw, how flaw information is provided (corrective actions and guidance on corrective actions). The procedures ensure that all reported flaws are corrected and that corrections are issues to TOE users, and that the flaw corrections do not introduce new flaws. The procedures also ensure a timely response to reported flaws and the automatic distribution of security flaw reports to the affected users

These activities are documented in:

- Supplement Guide for Life Cycle Support
- The SPR Process User Guide

- Software Problem Policy Information
- Sample Software Problem Report

The Life cycle support assurance measure satisfies the following EAL 2 augmented with ALC_FLR.1 assurance requirements:

- ALC_FLR.1

6.2.5 Tests

The Test Documentation is found in the following documents:

- UNICOS/mp 2.4 Security Test Plan
- UNICOS/mp 2.4 Security Test Summary Report, OS Test Group – Cray Inc., EAL2+ Security Function Verification
- Supplement Guide for Testing
- Test Source Code
- Actual Test Results

These documents describe the overall test plan, testing procedures, the tests themselves, including expected and actual results. In addition, these documents describe how the functional specification has been appropriately tested.

The Tests assurance measure satisfies the following EAL 2 augmented with ALC_FLR.1 assurance requirements:

- ATE_COV.1
- ATE_FUN.1
- ATE_IND.2

6.2.6 Vulnerability assessment

Cray performs vulnerability analyses of the TOE to identify weaknesses that can be exploited in the TOE. All of the SOF claims are based on password space calculations and based on the SOF rationale in this ST; a separate SOF analysis is not applicable.

These activities are documented in:

- Cray UNICOS/mp Operating System Version 2.4.15 on Cray X1 hardware Vulnerability Analysis

The Vulnerability assessment assurance measure satisfies the following EAL 2 augmented with ALC_FLR.1 assurance requirements:

- AVA_SOF.1
- AVA_VLA.1

7. Protection Profile Claims

This TOE does not claim conformance to a Protection Profile.

8. Rationale

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

- Security Objectives
- Security Functional Requirements
- Security Assurance Requirements
- TOE Summary Specification
- Security Functional Requirement Dependencies
- Internal Consistency

8.1 Security Objectives Rationale

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

8.1.1 Security Objectives Rationale for the TOE and Environment

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

	O.AUTHORIZATION	O.DISCRETIONARY_ACCESS	O.ENFORCEMENT	O.MANAGE	O.INSTALL	O.PHYSICAL	O.CREDEN	O.STAFF	O.PEER
P.AUTHORIZED_USERS	X		X	X					
P.NEED_TO_KNOW		X	X	X					
A.LOCATE						X			
A.MANAGE								X	
A.NO_EVIL_ADM					X		X		
A.PEER									X
A.PROTECT						X			

Table 3: Environment to Objective Correspondence

8.1.1.1 P.AUTHORIZED_USERS

Only those users who have been authorized to access the information within the system may access the system.

This policy is implemented by the O.AUTHORIZATION objective. The O.MANAGE supports this policy by requiring authorized administrators to be able to manage the functions and O.ENFORCEMENT ensures that functions are invoked and operate correctly.

8.1.1.2 P.NEED_TO_KNOW

The system must limit the access to, modification of, and destruction of the information in protected resources to those authorized users who have a 'need to know' for that information.

This policy is implemented by the O.DISCRETIONARY_ACCESS objective. The O.MANAGE supports this policy by requiring the authorized administrator is able to manage the functions and O.ENFORCEMENT ensures that functions are invoked and operate correctly.

8.1.1.3 A.LOCATE

The processing resources of the TOE will be located within controlled access facilities which will prevent unauthorized physical access.

This is addressed by O.PHYSICAL which addresses those parts of the TOE which are critical to security policy are protected from physical attack.

8.1.1.4 A.MANAGE

There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.

This is addressed by O.STAFF, which ensures that the TOE is managed by a competent staff.

8.1.1.5 A.NO_EVIL_ADM

The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the administrator documentation.

This is addressed by O.INSTALL, which ensures that the TOE is delivered, installed, managed and operated in a manner that maintains IT security. This is further addresses by O.CREDEN by ensuring that authentication data is protected.

8.1.1.6 A.PEER

Any other systems with which the TOE communicates are assumed to be under the same management control and operate under the same security policy constraints.

The O.PEER supports this policy by requiring all systems with which the TOE communicates be under the same management control and operate under the same security policy constraints.

8.1.1.7 A.PROTECT

The TOE hardware and software critical to security policy enforcement will be protected from unauthorized physical modification.

This is addressed by O.PHYSICAL which addresses those parts of the TOE which are critical to security policy are protected from physical attack.

8.2 Security Requirements Rationale

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

8.2.1 Security Functional Requirements Rationale

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

	FDP_ACC.1	FDP_ACF.1	FIA_ATD.1	FIA_UAU.2	FIA_UID.2	FIA_USB.1	FMT_MSA.1	FMT_MSA.3	FMT_MTD.1a	FMT_MTD.1b	FMT_MTD.1c	FMT_SMF.1	FMT_SMR.1	FPT_RVM.1	FPT_SEP.1
O.AUTHORIZATION			X	X	X					X					
O.DISCRETIONARY_ACCESS	X	X	X			X	X								
O.ENFORCEMENT														X	X
O.MANAGE								X	X	X	X	X	X		

Table 4: Objective to Requirement Correspondence

8.2.1.1 O.AUTHORIZATION

The TSF must ensure that only authorized users gain access to the TOE and its resources.

This TOE Security Objective is satisfied by ensuring that:

- FIA_ATD.1, FMT_MTD.1b: These security requirements provide authentication data and attributes and ensure that the authentication data is protected from modification from unauthorized users
- FIA_UID.2, FIA_UAU.2: These security requirements require a user to be identified and authenticated before any other TSF-mediation action on their behalf is allowed thereby ensuring only authorized users gain access to the TOE and its resources.

8.2.1.2 O.DISCRETIONARY_ACCESS

The TSF must control access to resources based on identity of users. The TSF must allow authorized users to specify which users may access which resources.

This TOE Security Objective is satisfied by ensuring that:

- FDP_ACC.1: The DAC policy covers the subjects and objects that are protected.
- FDP_ACF.1: Access to objects is based upon the security attributes; rules define access between subjects and objects.
- FIA_ATD.1, FIA_USB.1: Control of access to resources is based on the user identity defined and bound by these requirements.
- FMT_MSA.1: Authorized users must be able to control who has access to objects.

8.2.1.3 O.ENFORCEMENT

The TSF must be designed and implemented in a manner that ensures the organizational policies are enforced in the target environment.

This TOE Security Objective is satisfied by ensuring that:

- FPT_RVM.1 The TSF must make and enforce the decisions of the TSP.

- FPT_SEP.1: The TOE must be protected from interference that would prevent it from performing its functions.

8.2.1.4 O.MANAGE

The TSF must provide all the functions and facilities necessary to support the authorized administrators who are responsible for the management of TOE security.

This TOE Security Objective is satisfied by ensuring that:

- FMT_MTD.1a, FMT_MTD.1b, FMT_MTD.1c, FMT_SMF.1: The administrator must be able to administer user accounts and user's can change their own authentication data.
- FMT_MSA.3, FMT_SMR.1: The TSF must provide for an authorized administrator and an object creator to manage the TOE who can set and override default user settings.
- FMT_MSA.1: Ensures that the authorized administrator, (root) is able to modify the object security attributes.

8.3 Security Assurance Requirements Rationale

This ST contains the assurance requirements from the CC EAL assurance package augmented with the additional assurance Flaw Remediation [ALC_FLR]. The assurance requirements for the TOE are based on good commercial development practices. This ST has been developed for a specialized controlled environment where all users with access are equally trustworthy. The security environment provides restrictive physical protection. All peripheral devices reside within controlled access facilities. Communication paths are adequately protected. All other systems with which the TOE communicates are assumed as trustworthy as is the TOE. With these restrictions placed on the TOE, a moderate level of risk exists to the assets. As such, it is believed that EAL 2+ provides an appropriate level of assurance in the security functions offered by the TOE.

The only security mechanism that is realized by a probabilistic or permutational implementation is the password mechanism. For an analysis of the Strength of Function refer to Section 8.8.

8.4 Requirement Dependency Rationale

The following table shows the security functional requirement dependencies that exist based on the security functional requirements included in this Security Target. As indicated in the following table all of the dependencies are satisfied.

	FDP_ACC.1	FDP_ACF.1	FIA_ATD.1	FIA_UAU.1	FIA_UID.1**	FMT_MSA.1	FMT_MSA.3	FMT_SMF.1	FMT_SMR.1
FDP_ACC.1		X							
FDP_ACF.1	X						X		
FIA_ATD.1									
FIA_UAU.2					X				
FIA_UID.2									
FIA_USB.1			X						
FMT_MSA.1	X							X	X
FMT_MSA.3						X			X
FMT_MTD.1a								X	X
FMT_MTD.1b								X	X

	FDP_ACC.1	FDP_ACF.1	FIA_ATD.1	FIA_UAU.1	FIA_UID.1**	FMT_MSA.1	FMT_MSA.3	FMT_SMF.1	FMT_SMR.1
FMT_MTD.1c								X	X
FMT_SMF.1									
FMT_SMR.1					X				
FPT_RVM.1									
FPT_SEP.1									

Table 5 Dependency Analysis

Note: FIA_UAU.2 is dependent on FIA_UID.1. The TOE meets the SFR FIA_UID.2 that is hierarchal to FIA_UID.1; therefore this dependency is satisfied.

This Security Target also includes security assurance requirements that have dependencies. Since EAL 2 has been adopted in this Security Target and EAL 2 is defined in the Common Criteria, it is assumed that all of the dependencies within that assurance level have been addressed. The only change to the set of EAL 2 security assurance requirements made in this Security Target is that additional of ALC_FLR.1. ALC_FLR.1 has no dependencies and therefore all of the security assurance requirement dependencies are fulfilled.

8.5 Explicitly Stated Requirements Rationale

This Security Target does not contain any explicitly stated requirements. Note specifically that no U.S. National interpretations have been applied and International Interpretations have been identified where applied, but are not considered explicitly stated requirements.

8.6 TOE Summary Specification Rationale

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

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

	FDP_ACC.1	**FDP_ACF.1**	**FIA_ATD.1**	**FIA_UAU.2**	**FIA_UID.2**	**FIA_USB.1**	**FMT_MSA.1**	**FMT_MSA.3**	**FMT_MTD.1a**	**FMT_MTD.1b**	**FMT_MTD.1c**	**FMT_SMF.1**	**FMT_SMR.1**	**FPT_RVM.1**	**FPT_SEP.1**
User data protection	X	X													
Identification and authentication			X	X	X	X									
Security management							X	X	X	X	X	X	X		

Protection of the TSF														X	X

Table 6: Security Functions vs. Requirements Mapping

8.7 Strength of Function Rationale

The claimed SOF-medium level is primarily intended to be commensurate with the relative assurance afforded by EAL2. This security target only includes one probabilistic or permutational function. "Identification and Authentication" is the only relevant security functions and the security functional requirements include:

- FIA_UAU.2 – User authentication before any action

To SOF calculation is included in the ST and not further discussed in additional documentation. Assumptions from this ST are:

1. There are 72 possible characters
2. Each password must have at least six characters.
3. Each password must contain at least two alphabetic characters and at least one numeric or special character. In this case, ``alphabetic" means upper and lower case letters. Numeric or special character means any digit or special character on a standard keyboard above a digit.
4. Each password must differ from the user's login name and any reverse or circular shift of that login name. For comparison purposes, an upper case letter and its corresponding lower case letter are equivalent.
5. New passwords must differ from the old by at least three characters. For comparison purposes, an upper case letter and its corresponding lower case letter are equivalent.
6. A layman could make a password guess "by hand" every 5 seconds, and
7. A proficient or expert person could create a program to guess passwords.

There is one authentication mechanism associated with FIA_UAU.2. The default password is six alpha characters long and is case sensitive.

Assuming the most vulnerable scenario in which a user chooses a password with the smallest length (6 characters), the number of passwords that can be generated is given by

N^K,

where N is the number of characters available in the character set, and K is the number of characters used in the password.

Enforcing the password restrictions identified above limits the password space to

Passwords = 52*52*20*72*72*72 = 20,185,251,840

Enforcing the password restrictions identified in item 4 with a 6-character login name reduces the number by 6 shifts of the login name and 1 reversal or $2^6 * 7 = 448$

Total passwords = (52*52*20*72*72*72) - $(2^6 * 7)$ = **20,185,251,392**

On average, an attacker would have to guess (20,185,251,392/ 2) = **10,092,625,696** passwords to guess the correct password.

Case 1: Layman guessing "by hand."

In this case, 5 seconds/(password guess) is required. The average total time to guess the correct password can be estimated by:

(10,092,625,696 passwords) * (5 seconds / password guess) * (1 hour / 3600 seconds) * (1 day / 24 hours) * (1 year / 365 days) = **1600 years**[1]

From the attack potential tables (sum = *), the corresponding attack potential is HIGH, and the SOF rating = HIGH.

Case 2: Expert person creating a program to guess (but not using any specialized equipment).
In this case,, the proficient person creates a program that can guess 1000 passwords per second or .0001 seconds per guess. The average total time to guess the correct password can be estimated by:

(10,092,625,696 passwords) * (.0001second / password guess) * (1 hour / 3600 seconds) * (1 day / 24 hours) = **11 days.**

The attack potential table yields the following:

- Elapsed Time = <1 month = 5
- Expertise = Expert = 4
- Knowledge of TOE = Public = 2
- Access to TOE = <1 month = 6
- Equipment = Standard = 2
- Total = 19

From the attack potential tables (sum =19), the corresponding attack potential is Moderate, and the SOF rating = Medium.

Conclusions

The Security Target made a strength of function claim of SOF-MEDIUM. From the two cases presented above, UNICOS/mp operating system meets or exceeds the SOF claim of SOF-MEDIUM .

8.8 PP Claims Rationale

This TOE does not claim conformance to a Protection Profile.

[1] The calculation * was derived since Cray does not consider an attack that takes 1600 years practical.