# National Information Assurance Partnership


TM

# Common Criteria Evaluation and Validation Scheme Validation Report

## Cray

## UNICOS/lc

## Version 2.1

**Report Number: VID10217-0007-VR**

**Dated: 2008-12-12**

**Version: 1.0**

**National Institute of Standards and Technology**
**Information Technology Laboratory**
**100 Bureau Drive**
**Gaithersburg, MD 20899**

**National Security Agency**
**Information Assurance Directorate**
**9800 Savage Road STE 6757**
**Fort George G. Meade, MD 20755-6757**

## ACKNOWLEDGEMENTS

**Validation Team**

**The Aerospace Corporation**

Columbia, MD

**Noblis**

Falls Church, VA

**atsec Information Security Corporation**

Austin, TX

## Table of Contents


**1. EXECUTIVE SUMMARY ..... 4**

**2. IDENTIFICATION ..... 4**

**3. SECURITY POLICY ..... 5**

3.1. DISCRETIONARY ACCESS CONTROL ..... 5
3.2. I&A ..... 6
3.3. AUDITING ..... 6
3.4. OBJECT REUSE ..... 7

**4. ASSUMPTIONS ..... 7**

4.1. USAGE ASSUMPTIONS ..... 7
4.2. CLARIFICATION OF SCOPE ..... 7

**5. ARCHITECTURAL INFORMATION ..... 7**

**6. DOCUMENTATION ..... 8**

**7. IT PRODUCT TESTING ..... 9**

7.1. SPONSOR TESTING ..... 9
7.2. EVALUATOR TESTING ..... 11

**8. EVALUATED CONFIGURATION ..... 13**

*8.1.1. Evaluated configuration ..... 13*

**9. RESULTS OF THE EVALUATION ..... 14**

**10. VALIDATOR COMMENTS ..... 14**

**11. SECURITY TARGET ..... 15**

**12. LIST OF ACRYONYMS ..... 16**

**13. BIBLIOGRAPHY ..... 17**

# 1. EXECUTIVE SUMMARY

This document is intended to assist the end-user of this product with determining the suitability of the product in their environment. End-users should review both the Security Target (ST) which is where specific security claims are made, and this Validation Report (VR) which describes how those security claims were evaluated.

This report documents the NIAP Validators' assessment of the evaluation of Cray UNICOS/lc Version 2.1. It presents the evaluation results, their justifications, and the conformance results. This validation report is not an endorsement of the IT product by any agency of the U.S. Government and no warranty of the IT product is either expressed or implied.

The evaluation was performed by the atsec Information Security Corporation, and was completed during November 2008. atsec Information Security Corporation is an approved NIAP Common Criteria Testing Laboratory (CCTL). The evaluation was conducted in accordance with the requirements of the Common Criteria for Information Technology Security Evaluation, version 2.3. The information in this report is largely derived from the Evaluation Technical Report (ETR) and associated test report, both written by the CCTL. The evaluation determined the product to be **Part 2 extended, Part 3 conformant**, and to meet the requirements of **EAL3 augmented by ALC_FLR.1**. Additionally, the TOE was shown to satisfy the requirements of the Controlled Access Protection Profile (CAPP), Version 1.d, 8 October 1999.

Cray UNICOS/lc is a general purpose, multi-user, multi-tasking Linux based operating system executing on the Cray XT4 and XT5 computer systems. It provides a platform for a variety of applications in the governmental and commercial environments.

The evaluation covers the Cray XT4 and Cray XT5 computer systems consisting of nodes running evaluated software components of the Cray UNICOS/lc Operating System. Multiple TOE systems may be connected in a network. The hardware platforms selected for the evaluation consist of machines which were available when the evaluation completed and which will remain available for a substantial period of time afterwards.

The validation team agrees that the CCTL presented appropriate rationales to support the Results of Evaluation presented in Section 4, and the Conclusions presented in Section 5, of the ETR. The validation team therefore concludes that the evaluation (and its PASS result) for Cray UNICOS/lc 2.1 is complete and correct.

# 2. IDENTIFICATION

The Common Criteria Evaluation and Validation Scheme (CCEVS) is a National Security Agency (NSA) effort to establish commercial facilities to perform trusted product evaluations. Under this program, security evaluations are conducted by commercial testing laboratories called Common Criteria Testing Laboratories (CCTLs) using the Common Evaluation Methodology (CEM) for

Evaluation Assurance Level (EAL) 1 through EAL 4 in accordance with National Voluntary Laboratory Assessment Program (NVLAP) accreditation granted by the National Institute of Standards and Technology (NIST).

The NIAP Validation Body assigns Validators to monitor the CCTLs to ensure quality and consistency across evaluations. Developers of information technology products desiring a security evaluation contract with a CCTL and pay a fee for their product's evaluation. Upon successful completion of the evaluation, the product is added to NIAP's Validated Products List.

Table 1 provides information needed to completely identify the product, including:

- The Target of Evaluation (TOE): the fully qualified identifier of the product as evaluated
- The Security Target (ST), describing the security features, claims, and assurances of the product
- The Protection Profile to which the product is conformant
- The conformance result of the evaluation
- The organizations and individuals participating in the evaluation

**Table 1: Evaluation Identifiers**

| Item | Identifier |
|---|---|
| Evaluation Scheme | United States NIAP Common Criteria Evaluation and Validation Scheme |
| Target of Evaluation | Cray UNICOS/lc Operating System 2.1 |
| Protection Profile | Controlled Access Protection Profile (CAPP), Issue 1.d, 8 October 1999. |
| Security Target | *Cray UNICOS/lc Operating System 2.1 Security Target for CAPP compliance;* Version 1.15, 2008-10-07 |
| Evaluation Technical Report | *Evaluation Technical Report for a Target of Evaluation: Cray UNICOS/lc 2.1* Version 2, 2008-11-18 |
| Conformance Result | CC V2.3, Part 2 extended, Part 3 conformant, EAL 3 augmented by ALC_FLR.1, and CAPP-compliant |
| Sponsor | Cray, Inc. |
| Developer | Cray, Inc. |
| Evaluators | atsec information security corporation |
| Validators | The Aerospace Corporation |

# 3. SECURITY POLICY

## 3.1. Discretionary Access Control

Cray UNICOS/lc implements Discretionary Access Control (DAC) through the use of standard UNIX permission bits and the POSIX standard Access Control Lists (ACLs). A Discretionary Access Control policy requires mechanisms whereby the access of users (i.e., subjects) to system

resources and data (i.e., objects) can be controlled on the basis of user identity, role, and explicit permissions. Mechanisms that implement a DAC policy provide the capability for users to specify the how their personal data objects are to be shared.

Permission bits are associated with objects and specify the permissions (typically, READ, WRITE, EXECUTE) for a specific user, the user's group affiliation, and all others (i.e., "world"). Access Control Lists provide the same functionality relative to granting specific permissions, but are considerably more flexible in that they can identify a number of group affiliations for a single user.

The standard UNIX DAC mechanism is permission bits, as is the case with Cray UNICOS/lc. However, Cray UNICOS/lc implements ACLs as an extended permission mechanism, available at the discretion of the file owner; ACLs are supported only for file system objects on ext3 file systems.

## 3.2. I&A

Each user must have a unique identity (i.e., username plus password), and be authenticated prior to obtaining resources and services from the TOE. Note, however, that in a networked environment, user identities are unique to a server, and are neither known globally nor are universally unique. That is, each Cray computer system maintains its own set of users and their associated passwords and attributes. A user that has access to more than one server on a network will have a different user identity, and possibly different attributes, on each server for which access is authorized.

An administrator can define the following constraints for the authentication process:

- Maximum duration of a password (i.e., time-to-live)
- Minimum time allowed between password changes
- Minimum password length
- Number of days warnings are displayed prior to password expiration
- Allowed number of consecutive unsuccessful login attempts
- Disallowed passwords (i.e., the TOE retains a history of recently-used passwords to prevent users from cycling previously-used passwords)

The proper parameters for each of these choices is defined for the evaluated configuration.

## 3.3. Auditing

The TOE audit mechanism allows the generation of audit records for security-related events, and allows the administrator to configure the audit mechanism to collect which events are to be captured and which users are to be audited; it is also possible for the administrator to identify specific users that are not to be audited.

Each audit record contains event-specific information, and identifies whether the request that caused the event was successful or failed, and. An audit record consists of a standard header that includes the following information:

- A unique audit identifier

- The LoginID of the user who caused the audit record to be generated
- The Effective User ID of the user at the time the record was generated
- Date and time the audit record was generated
- Type of event

Audit records are stored in ASCII format, and can be searched through the use of the standard UNIX/LINUX *grep* tool.

### 3.4. Object Reuse

Although the TOE supports several different types of objects, each is managed by the system such that no pre-existing content is provided to users to whom objects are allocated. That is, whenever an object (e.g., buffers, memory extents, disk space) is allocated to a user process, it is managed such that any data that had previously been in the object (i.e., from an earlier process) is unavailable to the new process.

In short, memory pages are initialized to all zeroes when allocated to a process, IPC objects are also initialized to all zeroes, file system objects are created with no content (with the exception of directories and symbolic links).

## 4. ASSUMPTIONS

### 4.1. Usage Assumptions

Although there are several assumptions stated in the Security Target, the primary conditions are that:

- The TOE is located within controlled facilities and is protected from unauthorized physical access
- TOE hardware and software are protected from unauthorized modification
- All authorized users possess authorization for at least some of the data managed on the TOE
- The TOE operates in a relatively benign environment
- Unencrypted communications paths, and communications paths within the controlled facility are protected from unauthorized physical access

### 4.2. Clarification of Scope

The TOE includes the hardware platform (see Section 8) and all the code that enforces the policies identified (see Section 3). The TOE also includes secure communications functions; i.e., SSH V2 and SSL V3).

## 5. ARCHITECTURAL INFORMATION

The TOE is a multi-user, multi-tasking operating system which can support multiple users simultaneously. A fundamental protection mechanism is the memory management and virtual

memory support provided by the hardware. This provides a domain (i.e., supervisor state) in which only the kernel executes.

The TSF comprises two major components: kernel software and trusted processes. One instance of the kernel and the trusted processes execute on each node part of the Cray computer system hosting the operating system.

The kernel software executes in supervisor state, which is supported by the memory management mechanism in the hardware. The memory management mechanism insures that only kernel code can execute in the supervisor state (wherein all memory may be accessed), and also serves to protect the kernel code from external tampering. The kernel implements file and I/O services, which provides access to files and devices. The kernel also implements:

- Named pipes
- Unnamed pipes
- Signals
- Semaphores
- Shared memory
- Message queues
- Internet domain sockets
- Unix domain sockets

The trusted processes, which provide the remainder of the TSF, are referred to as “non-kernel TSF” services because they run in user state; they execute in the same hardware domain as user applications. These are protected from external tampering through the process management and memory virtualization mechanisms that implement per-process address spaces, which prevent processes from interfering with each other. They are also protected from unauthorized access by the access control mechanisms of the TSF. The primary non-kernel TSF services are:

- Identification and authentication
- Network application layer services
- Configuration and management commands requiring root privileges

# 6. DOCUMENTATION

The TOE is delivered with a combination of hardware and software specific documentation. Hardware specific documentation varies with the model of the TOE. The following software documentation is uniform across TOE hardware platforms:

- Common Criteria EAL3+ Evaluated Configuration Guide for Cray Unicos 2.1 v1.5 2008-08013
- Command references for the applications and configuration files implementing security functionality are available as man pages on an installed system

Additional guidance documents are available from Cray which have not been assessed by the evaluation. The above mentioned Evaluated Configuration Guide fully and completely explains how to install, configure, and administrate the TOE. Moreover, it provides explanations about the intended environment.

Additional man pages to the ones mentioned above are present on the system for applications, configuration files, APIs, and others which do not implement security functionality. These man pages have not been reviewed during the evaluation.

# 7. IT PRODUCT TESTING

## 7.1. Sponsor Testing

Test configuration

The test results provided by the developer were generated on the Cray XT4 and XT5 systems running UNICOS/lc 2.1 with associated:

- boot node
- SDB node
- login node
- other service nodes
- compute nodes

The developer has performed his tests on the above-listed hardware platforms. The software was installed and configured as defined in the Evaluated Configuration Guide with additional software packages identified in the Test Plan. The Test Plan presents the argument that those additional packages are within the boundary defined by the Security Target and do not constitute a violation of the evaluated configuration (see the chapter headed “Target of Evaluation (TOE) compliance” in the Test Plan).

Testing approach

The Test Plan provided by the developer lists test cases by groups, which reflects the mix of sources for the test cases. The mapping provided lists the TSF/TSFI with which the test cases are associated. The Test Plan is focused on the security functions of the TOE and ignores other aspects typically found in developer test plans. The test cases are mapped to the corresponding Functional Specification and HLD.

The developer uses several test suites that are integrated into one test system that includes automatic and manual tests to test the TOE.

The LTP test suite is an adapted version of tests from the Linux Testing Project. The LTP tests have a common framework in which individual test cases adhere to a common structure for setup, execution, and cleanup of tests. Each test case may contain several tests of the same function, stressing different parts (for example, base functionality, behavior with illegal parameters, and

reaction to missing privileges). Each test within a test case reports PASS (or, OK) or FAIL, and the test case summary in batch mode reports PASS if all the tests within the test case passed, otherwise FAIL.

The ACL tests are structured in a very specific way. The test cases are comments, shell scripts, and expected output. The driver script for the test cases runs the shell commands and compares the output with the expected output in the test scripts. Each output line that matches is tagged with OK, each line that does not match is tagged with FAILED. The driver scripts summarize the OK/FAILED entries and report the number of each of the two flags at the end. The test case reports 101 OK entries when executed successfully. The tests are started in batch mode via the *runme* shell script.

The OpenSSL tests execute a part of the LTP OpenSSL test suite adapted for the security evaluation.

The audit tests use their own testing framework, where each test is executed twice: once with a positive test goal and once with a negative test goal. The audit tests that do not cover system calls directly, but cover the supporting tools, use a similar approach of iterating over the various stages as far as applicable. For each of the areas in the audit test suite, a driver program will perform global setup and run the individual test cases. Results are collected into the log file showing PASS or FAIL verdicts.

The manual tests cover functionality that cannot easily be tested in an automated way, such as serial terminals.

The test results of the developer can be found in [TRES]. All the tests were executed successfully (PASS/OK) apart from the test cases that are documented to fail or be skipped in the [TP]. The test systems were configured according to the ST and Cray internal setup instructions to achieve the evaluated configuration. The manual test results included in [TRES] also include PASS/FAIL labeling by the developer.

The test results provided by the developer were generated on the following above mentioned systems.

Testing results

The test results provided by the developer were generated on the hardware platforms listed above. As described in the testing approach, the test results of all the automated tests are written to files. In addition, a log file for the LTP tests reports more details on the flow of the tests.

The test results of the few manual tests performed have been recorded by the developer, and those results have been presented in separate files.

All test results from all tested platforms show that the expected test results are identical to the actual test results, considering the expected failures stated in the developer’s test plan.

Test coverage

The functional specification has identified the following TSFI:

- system calls
- security-critical configuration files (TSF databases)

- trusted programs
- network applications
- virtual files

A mapping provided by the developer shows that the tests cover all individual TSFI identified for the TOE. An extension to this mapping developed by the evaluator as documented in the test case coverage analysis document shows that significant details of the TSFI have also been tested with the developer's test suite. Therefore, the requirements for the evaluation have been satisfied, since an exhaustive interface specification testing is not required.

Test depth

In addition to the mapping to the functional specification, the developer provided a mapping of test cases to subsystems of the high-level design and the internal interfaces described in the high-level design. This mapping shows that all subsystems and the internal interfaces are covered by test cases. To show evidence that the internal interfaces have been called, the developer provided the description of the internal interfaces as part of the high-level design. The interfaces are clear enough to allow the evaluator to assess whether they have been covered by testing.

Not all of the internal interfaces mentioned in the high-level design could be covered by direct test cases. Some internal interfaces can – due to the restrictions of the evaluated configuration – only be invoked during system startup. This especially includes internal interfaces to load and unload kernel modules, to register /deregister device drivers, and install / deinstall the interrupt handler. Since the evaluated configuration does not allow dynamic loading and unloading of device drivers as kernel modules, those interfaces are only used during system startup and therefore, are implicitly tested there.

Conclusions

The evaluator has verified that developer testing was performed on hardware conformant to the ST.

The evaluator was able to able to follow and fully understand the developer's testing approach by using the information provided by the developer.

The evaluator analyzed the developer testing coverage and the depth of the testing by reviewing all test cases as shown in [TCA]. The evaluator found the testing of the TSF to be extensive and covering the TSFI as identified in the functional specification.

The evaluator reviewed the test results provided by the developer and found them to be consistent with the test plan.

## 7.2. Evaluator Testing

TOE test configuration

The evaluator verified the correct installation and configuration of the test systems according to the documentation in the Evaluated Configuration Guide [ECG] and the test plan.

Cray XT4 computer system:

The XT4 computer system was located at the developer's facility in Chippewa Falls, WI. The exact hardware and software configuration of the test system is described in ATE_FUN.1-4, as this machine is the same that was used by the developer for his testing, except that at the time of the evaluator testing, the XT5 blade was not installed.

Subset size chosen

As the evaluator was integrated into the developer's test team during the development of the test cases and also has acquired knowledge about the LTP test suite from previous evaluations, the evaluator chose to observe the developer test team executing the tests, rather than executing the tests himself. The evaluator observed the test execution while he was at Mendota Heights, MN during his on-site visit. In addition, the evaluator had remote access to the test machine and was, therefore, able to observe the developer while testing was performed.

Evaluator tests performed

In addition to repeating all the automated developer tests, the evaluator devised tests for a subset of the TOE. The tests are listed in the Evaluator Test Plan [TPE].

The evaluator chose these tests for the following reasons:

- The test cases examine some of the security functions of the TOE in more detail than the sponsor supplied test cases (object reuse).
- The test cases cover aspects not included in the developer testing (access enforcement on read/write of a file, validation of proper system configuration).
- Because the developer-supplied test cases already cover the TOE in a broad sense, the evaluator has devised only a small set of test cases.

The evaluator created several test cases for testing a few functional aspects where the developer test cases were not considered by the evaluator to be broad enough. During the evaluator coverage analysis of the test cases provided by the developer, the evaluator gained confidence in the developer testing effort and the depth of test coverage in the developer -supplied test cases.

Summary of Evaluator test results

The evaluator testing effort consists of two parts. The first is the observation of the developer test cases, and the second is the execution of the tests created by the evaluator.

The tests were performed on the system located within the developer's facility in Chippewa Falls, WI. The system was attached to the evaluator's laptop via the local Ethernet cabling. The UNICOS/lc 2.1 Server operating system with the test cases and test tools were installed on the machine. As the system was preconfigured, the evaluator verified different aspects of the configuration to ensure that the system was in the evaluated configuration. The test system was, therefore, configured according to the [ST].

The evaluator executed the above-mentioned tests. All the test results conformed to the expected test results from the test plan.

In addition to running the tests that were provided by the developer according to the test plan [TPE], the evaluator decided to run some additional test cases on the provided test systems:

- Permission settings of relevant configuration files
  This test was performed to verify that important configuration files had appropriate permission settings, considering the fact that on the nodes accessible to users, the file system is mounted read-only.
- Permission settings of applications
  This test was performed to verify that applications' configuration files had appropriate permission settings with respect to SUID and SGID bits, considering the fact that on the nodes accessible to users, the file system is mounted read-only.
- Permission settings on devices
  This test was performed to verify that applications' configuration files had appropriate permission settings with respect to SUID and SGID bits, considering the fact users who can access device files have direct access to the underlying resource.
- Verification of the use of MD5 passwords
  This test verified that passwords with a size of more than 8 characters can be used.
- Verification the SUID programs do not change the real UID
  This test was performed to verify that SUID programs do not change the real UID, only the effective UID.
- Testing of object reuse in regular file system objects
  This test checks for object reuse in regular files by creating a large spares file and trying to find non-zero data in the spares area.
- Test for access check enforcement
  This test verifies that the decision made during open() system call (e.g., open the file as read-only) is enforced during read/write system calls.
- Test for LD_LIBRARY_PATH
  This test checks that the environment variable LD_LIBRARY_PATH is unset when calling a SUID/SGID application.
- Test for open network ports
  This test verified that only trusted applications can open network ports.

All tests passed successfully.

# 8. EVALUATED CONFIGURATION[1]

### 8.1.1. Evaluated configuration

The Security Target defines the following hardware basis for the TOE:

[1] For more complete information on the evaluated configurations, see Section 1.5 of the Security Target.

- Cray XT4 computer system
- Cray XT5 computer system, without the XT5h blade using a vector-based CPU

The following types of nodes are supported as part of the TOE:

- Service nodes:
    - Login nodes available for interactive login by non-administrative users.
    - Other service nodes with access restricted to administrators, for example the servers for the Lustre and NFS filesystems.
- Compute nodes running a limited subset of the TOE software that are available for processes invoked by non-administrative users through ALPS.

The Security Target also defines the system configuration to include IPv4, the ext3/Lustre/NFSv3 file system, the ISO 9660 file system for CD-ROM, and various other virtual file systems.

# 9. RESULTS OF THE EVALUATION[2]

The evaluation team determined the product to be **CC Part 2 extended, CC Part 3 conformant, CAPP conformant,** and to meet the requirements of **EAL 3 augmented by ALC_FLR.1**. In short, the product satisfies the security technical requirements specified in *Cray UNICOS/lc Operating System 2.1 Security Target for CAPP compliance*, Version 1.15, 2008-10-07.

# 10. VALIDATOR COMMENTS

The Validator determined that the evaluation and all of its activities were performed in accordance with the CC, CEM, and CCEVS practices.

The Validator has the following observations:

- While the TOE distribution media includes a Graphical User Interface (GUI), it is not installed by default, is not part of the Evaluated Configuration and was not evaluated.

- The end-user should be aware of the following residual vulnerabilities and determine if they will have an impact in the expected operational environment:
    - CVE-2008-3077 – Denial of Service (DoS); only affects single nodes, the only node that may be significantly impacted is the job database node, which is only accessed by trusted administrators. Compute and login nodes would require a reboot, which is a limited denial of service.
    - CVE-2008-2931 – This only allows modification a mount flag, not the mount itself.
    - CVE-2008-2365 – DoS; only affects single nodes, the only node that may be significantly impacted is the job database node, which is only accessed by trusted

[2] The terminology in this section is defined in CC Interpretation 008, specifying new language for CC Part 1, section/Clause 5.4.

administrators. Compute and login nodes would require a reboot, which is a limited denial of service.

- CVE-2008-0891 – This vulnerability would require a restart of stunnel if successfully exploited.

# 11. SECURITY TARGET

The ST, *Cray UNICOS/lc Operating System 2.1 Security Target for CAPP compliance*, Version 1.15, 2008-10-07 is included here by reference.

# 12. LIST OF ACRYONYMS

| | |
|---|---|
| CC | Common Criteria |
| CCEVS | Common Criteria Evaluation and Validation Scheme |
| CCTL | Common Evaluation Testing Laboratory |
| CEM | Common Evaluation Methodology |
| EAL | Evaluation Assurance Level |
| ETR | Evaluation Technical Report |
| NIAP | National Information Assurance Partnership |
| NIST | National Institute of Standards & Technology |
| NSA | National Security Agency |
| PP | Protection Profile |
| ST | Security Target |
| TOE | Target of Evaluation |
| TSF | TOE Security Function |
| TSFI | TOE Security Function Interface |

# 13. BIBLIOGRAPHY

[1] Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, Version 2.3.

[2] Common Criteria for Information Technology Security Evaluation – Part 2: Security functional requirements, Version 2.3.

[3] Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance requirements, Version 2.3.

[4] Common Evaluation Methodology for Information Technology Security – Part 1: Introduction and general model, Version 2.3.

[5] Common Evaluation Methodology for Information Technology Security – Part 2: Evaluation Methodology, Version 2.3.