National Information Assurance Partnership

Common Criteria Evaluation and Validation Scheme Validation Report

IBM

SUSE LINUX Enterprise Server

Version 10 Service Pack 1

Report Number: CCEVS-VR-VID10271-2007

Dated: 2007-10-08

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 6740
Fort George G. Meade, MD 20755-6740

ACKNOWLEDGEMENTS

Validation Team

The Aerospace Corporation

Columbia, MD

Noblis

Falls Church, VA

atsec Information Security Corporation

Austin, TX

Table of Contents

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 Validation Report documents the NIAP validators' assessment of the evaluation of IBM SUSE LINUX Enterprise Server (SLES) Version 10 Service Pack 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 in the United States, and was completed during July 2007. 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 **Evaluation Assurance Level 4 (EAL4) augmented by ALC_FLR.3**. Additionally, the TOE was shown to satisfy the requirements of the **Controlled Access Protection Profile (CAPP), Version 1.d, 8 October 1999.**

SUSE LINUX Enterprise Server is a general purpose, multi-user, multi-tasking Linux based operating system. It provides a platform for a variety of applications in the governmental and commercial environment. SUSE LINUX Enterprise Server is available on a broad range of computer systems, ranging from departmental servers to multi-processor enterprise servers and small server type computer systems.

The SUSE LINUX Enterprise Server evaluation covers a potentially distributed, but closed network of IBM System x, System p5, System z, and eServer servers running the evaluated versions and configurations of SUSE LINUX Enterprise Server. The hardware platforms selected for the evaluation consist of machines which are available when the evaluation has completed and to 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 the Pass results for SUSE LINUX Enterprise Server 10 SP1 are complete and correct. Section 10 of this document should be reviewed by the end-user for specific Validator comments.

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 National Information Assurance Partnership (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 conformance result of the evaluation, the Evaluation Technical Report or ETR;
- The Protection Profile to which the product is conformant;
- 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	IBM SUSE LINUX Enterprise Server 10 SP1
Protection Profile	Controlled Access Protection Profile (CAPP), Issue 1.d, 8 October 1999.
Security Target	*SUSE Linux Enterprise Server 10 SP1 Security Target for CAPP compliance;* Version 1.5, 8 October 2007
Evaluation Technical Report	*Evaluation Technical Report a Target of Evaluation: SUSE LINUX Enterprise Server 10 Service Pack 1* Version 1.0, 2 July 2007
Conformance Result	CC V2.3, Part 2 extended, Part 3 conformant, EAL 4 augmented by ALC_FLR.3, and CAPP-compliant
Sponsor	IBM
Developer	IBM and SUSE/Novell
Evaluators	atsec information security corporation
Validators	The Aerospace Corporation, Noblis

3. SECURITY POLICY

3.1. Discretionary Access Control

SUSE LINUX Enterprise Server 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 SLES. However, SLES implements ACLs as an extended permission mechanism, available at the discretion of the file owner; ACLs are supported only for file system objects.[1]

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 server 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.

Users can change their own passwords. However, 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

[1] See Section 6.2.4 of the ST for a fuller discussion of the DAC mechanisms and the algorithm by which access determinations are made.

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).[2]

3.5. Security Management

The TOE supports two roles; normal users and administrative users. The management of the security critical parameters of the TOE is performed by administrative users. A set of commands that require root privileges is used for system management. Security parameters are stored in specific files that are protected by the access control mechanisms of the TOE against unauthorized access by users that are not administrative users.

Normal users are only able to execute those security commands they have been granted access to, such as the ability to change their passwords.

[2] A more complete discussion of object reuse for each of the various object types is contained in Section 6.2.5 of the ST.

3.6. Secure Communication

The TOE supports secure communication with other systems via the SSH v2 and SSL v3 protocol. Communication via the SSH v2 and SSL v3 protocols is protected against unauthorized disclosure and modification via cryptographic mechanisms. The TOE also allows for secure authentication of the communicating parties using the SSL v3 protocol with client and server authentication. This allows establishing a secure communication channel between different machines running the TOE even over an insecure network. The SSL v3 protocol can be used to tunnel otherwise unprotected protocols in a way that allows an application to secure its TCP based communication with other servers (provided the protocol uses a single TCP port).[3]

3.7. Protection of TOE Security Functions

The TOE, including the hardware and firmware components, is required to be physically protected from unauthorized access. While in operation, the kernel software and data are protected by the hardware memory protection mechanisms. The memory and process management components of the kernel ensure a user process cannot access kernel storage or storage belonging to other processes.

Non-kernel TSF software and data are protected by DAC and process isolation mechanisms. In the evaluated configuration, the reserved user ID root owns the directories and files that define the TSF configuration. In general, files and directories containing internal TSF data (e.g., configuration files) are also protected from reading by DAC permissions. The system kernel mediates all access to hardware components that are protected from direct access by user programs.

4. ASSUMPTIONS

4.1. Usage Assumptions

Although there are several assumptions stated in the Security Target[4], 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.

[3] A complete list of permitted cipher suites is listed in the Security Target, Section 6.2.7.1.

[4] See section 3.4 of the ST

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). TOE also includes secure communications functions; i.e., SSH V2 and SSL V3). The cryptographic implementation was not evaluated by NIST but rather by a combination of vendor-assertion and evaluation by another independent entity.[5]

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.

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 that 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.

[5] Details of the third-party cryptographic evaluation appear in the Security Target.

6. DOCUMENTATION

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

- Common Criteria EAL4+ Evaluated Configuration Guide for SUSE Linux Enterprise Server on IBM Hardware v2.4 2007-07-02
- 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 SUSE 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 sponsor were generated on the following systems:

System z
Hardware: z900/z9
Host Operating system running: z/VM 5.1 or z/VM 5.3 within a PR/SM logical partition

Opteron
Hardware: model 3455, Bladecenter LS-21

System p
Hardware: p5 720 (9124), Bladecenter JS-21
Host system running: LPAR partition

System x 3550, HS-20 Bladecenter, HS-21 Bladecenter
Hardware: Intel Xeon with Hyperthreading and EM64T

The sponsor has performed their tests on the above listed hardware platforms. The software was installed and configured as defined in the Evaluated Configuration Guide [ECG] with additional software packages identified in the Test Plan [TP]. The Test Plan presents the arguments 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 [TP]).

The evaluator performed an analysis of the remainder of the hardware stated in the Security Target that has not been subject to testing. The evaluator verified that testing covers the different implementations of security functionality offered by the hardware (memory separation and memory access control). Since all different implementations have been tested, the remaining hardware systems do not differ in their security functionality.

It is to be noted that the sponsor develops all systems listed in the ST and that they perform compatibility tests of the hardware using Linux in a non-evaluated configuration. Although this testing is not subject to assessment in this evaluation (and therefore is not covered by the evaluation) the customer may gain further confidence about the operation of the hardware.

Testing approach

The Test Plan provided by the sponsor lists test cases by groups, which reflects the mix of sources for the test cases. The mapping provided lists the TSF/TSFI the test cases are associated with. 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 sponsor uses one test suite including manual tests to test the TOE. However, the test suite comprises several independent test suites taken from Open Source communities. These test suites have been enhanced by sponsor-written test cases. The following test suites are part of the overall test system.

The LTP test suite is an adapted version of tests from the Linux Testing Project of which the sponsor is a member. 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, OK or FAIL and the test case summary in batch mode reports PASS if all the tests within the test case passed, otherwise FAIL. This test suite has been enhanced with audit test cases.

Tests can be executed either manually by running the test case file or run in batch mode by executing *make run*. When the test cases are run individually no log summary is generated. The user running the test cases has to inspect *stdout* and *stderr* of the process.

The ACL tests are structured in a very particular 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 *ok* entries when executed successfully. The tests are started in batch mode.

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

The manual tests cover functionality that can not easily be tested in an automated way, such as console login. The manual test descriptions are accompanied by template text files that detail the exact steps required, along with the expected results. The tester creates a copy of the template, inserts the actual results, and compares them with the expected ones manually.

The test results were provided by the sponsor. All the tests were executed successfully (pass/ok). The test systems were configured according to the ST and the instructions in the Evaluated Configuration Guide. The manual test results also include PASS/FAIL labeling by the sponsor.

Testing results

The test results provided by the sponsor 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 have been recorded by the sponsor 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 test plan.

Test coverage

The functional specification has identified the following TOE Security Functional Interfaces (TSFI):

- system calls
- security critical configuration files (TSF databases)
- trusted programs
- virtual files
- network protocols

A mapping provided by the sponsor shows that the tests cover all individual TSFI identified for the TOE. An extension to this mapping developed by the evaluators as documented in the test case coverage analysis document show that also significant details of the TSFI have been tested with the sponsor's test suite. This therefore satisfies the requirements for the evaluation, since an exhaustive specification testing is not required as outlined in CEM, paragraph 1496.

Test depth

In addition to the verification of the internal interface coverage, the evaluator verified that all security functions are tested with an appropriate depth. This means that the evaluator verified that different aspects of a particular security function is tested.

In addition to the mapping to the functional specification, the sponsor provided a mapping of test cases to subsystems of the high level design. This mapping shows that all subsystems are covered by test cases. Using the high-level design, the coverage of internal interfaces was evident. To show evidence that the internal interfaces have been called, the sponsor provided the results of test cases that had been executed on a system installed and configured in compliance with the Security Target and the Evaluated Configuration Guide [ECG] but where large parts of the kernel had been compiled with the instrumentation for the *gcov* coverage analysis tool. This tool allows extracting a profile of all the source code statements that have been executed as part of the tests including also numbers showing how often each source code statement has been executed. Part of the depth analysis was based on the output generated with those *gcov* instrumented kernels.

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

7.2. Evaluator Testing

TOE test configuration

The evaluator independently installed the test systems according to the documentation in the Evaluated Configuration Guide [ECG] and the test plan. As assessed in the evaluation report on the administrator guidance, [ECG] is consistent with the ST. Hence, the evaluator concludes that the evaluator’s configuration is consistent with the ST.

System p:

The IBM System x 3455 is located at the sponsor labs in Austin, Texas. The hardware configuration is equivalent to the system used by the sponsor to perform testing (see ATE_FUN.1-12). The exact hardware and software configuration of the test system can be found in [TPE], Appendix A.1.

Subset size chosen

As the evaluator worked closely with the sponsor’s test team, he was able to “look the team over the shoulder” during their testing. The evaluator was able to observe the installation and test execution for the different platforms. Moreover, the evaluator was involved in discussions during the development of the TOE and the test cases. Thus, the evaluator is convinced that the test results from the developer are trustworthy.

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 has chosen 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, Audit data protection records, password quality)

- The test cases cover aspects not included in the developer testing (verification of the long password support, verification of the ACL support in the archival tool)
- As the sponsor-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 sponsor test cases were not considered by the evaluator to be broad enough. During the evaluator coverage analysis of the test cases provided by the sponsor, the evaluator gained confidence in the sponsor testing effort and the depth of test coverage in the sponsor supplied test cases. The analysis has shown a very wide coverage of the TSF, therefore the evaluator devised only a small number of test cases.

Summary of Evaluator test results

The evaluator testing effort consists of two parts. The first one is the observation of the sponsor's test approach and execution of test cases and the second is the execution of the tests created by the evaluator.

The tests were performed at the at the sponsor's facility in Austin, Texas. The systems available for testing are listed above.

The TOE as well as the test cases and test tools were installed on all machines by the evaluator according to the instructions in the Evaluated Configuration Guide which refers to a fully automated installation process to achieve the evaluated configuration and the evaluator's test plan.

The log files generated by the test cases were analyzed for completeness and failures. The sponsor provided the following groups of automated test cases: LTP test cases, ACL tests, at and cron tests, OpenSSL tests and miscellaneous tests.

The test cases from LTP generate a log file that lists FAIL/PASS for all the test cases contained in this group.

The cron test cases generate a log on stdout that contains a list of PASSED/FAILED test cases.

The ACL test cases report ok/failed and summarize the number of failed and passed test cases at the end.

The OpenSSL test cases produced test information on stdout that included PASS/FAIL lines.

The miscellaneous test cases produce test results that show the pass or fail verdict.

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 sponsor according to the test plan from the sponsor, the evaluator decided to run some additional test cases on the provided test systems:

- Password Quality Tests

  This test was performed to verify that the password quality settings prevent trivial passwords. See [TPE], section 3.1
- Verification of the use of MD5 passwords

This test was performed to verify that long passwords can be used on the TOE due to the MD5 algorithm used for storing the passwords instead of using the classic crypt function that truncates passwords at eight characters. See [TPE], section 3.2

- 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. See [TPE], section 3.3.

- 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. See [TPE], section 3.4.

- Check for data import / export with DAC enforcement

  Although no claims in the ST are made about data import and export, the evaluator deemed it necessary to check for the correct functioning of the star utility mentioned for this purpose in the Evaluated Configuration Guide. By testing this utility the evaluator also had a simple ACL enforcement test. See [TPE] section 3.5 for the test case.

- Disabling of PAM password proposal

  The evaluator verified whether the password proposal during password change can be disabled. See [TPE], section 3.6.

- Access check enforcement

  This test case demonstrates whether the access restrictions validated during open are enforced during read/write. See [TPE], section 3.7.

- LD_LIBRARY_PATH handling

  This test case demonstrates whether LD_LIBRARY_PATH environment variable is honored for SUID/SGID applications. See [TPE], section 3.8.

All tests passed successfully.

8. EVALUATED CONFIGURATION[6]

The evaluated configurations are:

- System x: x3550 (rack mount), HS20 and HS21 (blades)
- Opteron (AMD): x3455 (rack mount), LS21 (blade)
- System p: any POWER5/POWER5+ compliant system or software

[6] For more complete information on the evaluated configurations, see Section 2.4 of the Security Target.

- System z: any z/Architecture compliant system or software

9. RESULTS OF THE EVALUATION[7]

The evaluation team determined the product to be **CC** v2.3 **Part 2 extended, CC v2.3 Part 3 conformant, CAPP conformant,** and to meet the requirements of **EAL 4 augmented by ALC_FLR.3**. In short, the product satisfies the security technical requirements specified in *SUSE Linux Enterprise Server 10 SP1 Security Target for CAPP compliance*, Version 1.5, 2007-10-08.

10. VALIDATOR COMMENTS

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

The Validators have the following observations:

- A Graphical User Interface (GUI) is not part of the Evaluated Configuration and was not evaluated.
- Several residual vulnerabilities exist that do not affect TOE Security Functions. While most of these vulnerabilities have been addressed with updated code, those updates are not available in the evaluated version of the TOE and were not tested. It is the responsibility of the end-user to ensure that the TOE is not adversely affected by the following vulnerabilities:
  - CVE-2007-2878, type: Denial of Service
  - CVE-2007-2876, type: Denial of Service
  - CVE-2007-2172, type: Denial of Service
  - CVE-2007-1592, type: Denial of Service
  - CVE-2007-1496, type: Denial of Service
  - CVE-2007-1388, type: Denial of Service
  - CVE-2007-0006, type: Denial of Service

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

11. SECURITY TARGET

SUSE Linux Enterprise Server 10 SP1 Security Target for CAPP compliance, Version 1.5, 2007-10-08.

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
SMP	Symmetric Multiprocessing
TOE	Target of Evaluation
TSF	TOE Security Function
TSFI	TOE Security Function Interface
UP	Uniprocessor

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.