BSI-DSZ-CC-0143-1999

for

B1/ EST-X Version 2.0.1 with AIX, Version 4.3.1

from

Bull S.A. and IBM Informationssysteme Deutschland GmbH

Certification Report

BSI - Bundesamt für Sicherheit in der Informationstechnik, Postfach 20 03 63, D-53133 Bonn
Telefon +49 228 9582-0, Infoline +49 228 9582-111, Telefax +49 228 9582-455

Deutsches
IT-Sicherheitszertifikat

erteilt vom

Bundesamt für Sicherheit in der Informationstechnik

BSI

Bundesamt für Sicherheit
in der Informationstechnik

BSI-DSZ-CC-0143-1999

Operating System
B1/EST-X Version 2.0.1 with AIX Version 4.3.1

from

Bull S.A. and IBM Informationssysteme Deutschland GmbH

Common Criteria
Arrangement

The IT Product identified in this certificate has been evaluated at an accredited and licensed evaluation facility using the *Information Technology Security Evaluation Manual (ITSEM), Version 1.0* and the *Common Methodology for Information Technology Security Evaluation (CEM), Version 0.4,* for conformance to the *Common Criteria for Information Technology Security Evaluation (CC), Version 2.0.*

Evaluation Results:

Functionality:	**F-B1 of ITSEC conformant with Part 2 of CC**
Assurance Package:	**EAL 4 augmented by ALC_FLR.2 (Life Cycle Support – Flaw Reporting Procedures)**

This certificate applies only to the specific version and release of the product in its evaluated configuration and in conjunction with the complete Certification report.

The evaluation has been conducted in accordance with the provisions of the certification scheme of the Bundesamt für Sicherheit in der Informationstechnik and the conclusions of the evaluation facility in the evaluation technical report are consistent with the evidence adduced.

The remarks printed on the reverse side are part of this certificate.

Bonn, 11.03.1999

The President of the Bundesamt für
Sicherheit in der Informationstechnik

Dr. Henze L.S.

Bundesamt für Sicherheit in der Informationstechnik
Godesberger Allee 183 – D-53175 Bonn – Postfach 20 03 63 – D-53133 Bonn
Telefon +49 228 9582-0 – Telefax +49 228 9582-455 – Infoline +49 228 9582-111

This certificate is not an endorsement of the IT product by the Bundesamt für Sicherheit in der Informationstechnik or any other organisation that recognises or gives effect to this certificate, and no warranty of the IT product by the Bundesamt für Sicherheit in der Informationstechnik or any other organisation that recognises or gives effect to this certificate, is either expressed or implied.

Preliminary Remarks

Under the BSIG[1] Act, the Bundesamt für Sicherheit in der Informationstechnik (BSI) has the task of issuing certificates for information technology products.

A product is certified at the instigation of the vendor or a distributor, hereinafter called a sponsor.

A part of the procedure is the technical examination (evaluation) of the product according to the security criteria published by the BSI or generally recognised security criteria.

The evaluation is normally conducted by an evaluation facility recognised by BSI or by BSI itself.

The result of the certification procedure is the present certification report. This report includes, among others, the certificate (summarised assessment) and the detailed certification results.

The certification results contain the security description of the certified product, the details of the evaluation and stipulations regarding operation.

1 Act setting up the Bundesamtes für Sicherheit in der Informationstechnik (BSI-Errichtungsgesetz, BSIG) of 17 December 1990, Bundesgesetzblatt I p. 2834

Contents

A Certification

1 Specifications of the Certification Procedure

The certification body conducts the procedure according to the criteria laid down in the following:

- BSIG[2]
- BSI Certification Ordinance[3]
- BSI Schedule of Costs[4]
- Special decrees issued by the Bundesministerium des Innern (German Federal Ministry of the Interior)
- DIN EN 45011
- BSI Certification – Description of the Procedure (BSI 7125)
- Common Criteria for Information Technology Security Evaluation (CC), Version 2.0[5], May 1998
- Information Technology Security Evaluation Manual (ITSEM), Version 1.0[6], September 1993
- Draft of Common Methodology for Information Technology Security Evaluation (CEM), Version 0.4

---

2 Act setting up the Bundesamtes für Sicherheit in der Informationstechnik (BSI-Errichtungsgesetz, BSIG) of 17 December 1990, Bundesgesetzblatt I p. 2834

3 Ordinance on the Procedure for Issuance of a Certificate by the Bundesamt für Sicherheit in der Informationstechnik (BSI-Zertifizierungsverordnung, BSIZertV) of 7 July 1992, Bundesgesetzblatt I p. 1230

4 Schedule of Costs for Official Proceedings of Bundesamtes für Sicherheit in der Informationstechnik (BSI-Kostenverordnung, BSI-KostV) of 29 October 1992, Bundesgesetzblatt I p. 1838

5 Proclamation of the Bundesministeriums des Innern of 16 February 1999, in the Gemeinsames Ministerialblatt 1999 p. 1945.

6 Proclamation of the Bundesministeriums des Innern of 15 July 1992, in the Gemeinsames Ministerialblatt 1992 p. 546.

2 Recognition Agreements

In order to avoid multiple certification of the same product in different countries a mutual recognition of IT security certificates – as far as such certificates are based on ITSEC or CC - under certain conditions was agreed.

2.1 ITSEC - Certificates

An agreement on the mutual recognition of certificates based on ITSEC became effective on 3 March 1998. This agreement was signed by the national bodies of Finland, France, Germany, Greece, Italy, The Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and the United Kingdom.

2.2 CC - Certificates

An arrangement on the mutual recognition of certificates based on the CC evaluation assurance levels up to and including EAL 4 became effective on 5 October 1998 between the national certification bodies of France, Germany, United Kingdom, Canada and the United States.

3 Performance of Evaluation and Certification

The certification body monitors each individual evaluation to ensure a uniform procedure, a uniform interpretation of the criteria and uniform ratings.

The product B1/EST-X Version 2.0.1 with AIX Version 4.3.1 has undergone the certification procedure at BSI. It has been a re-certification based on the certificate No 97/81 for the product Bull BEST-X/B1, Version 1.1.1.9 and the certificate BSI-ITSEC-0138-1998 for the product AIX Version 4.3. The certificate No 97/81 was issued by the Certification Body of the UK IT Security Evaluation and Certification Scheme.

The re-evaluation of the product B1/EST-X Version 2.0.1 with AIX Version 4.3.1 was conducted by Department CC33 of Industrieanlagen-Betriebsgesellschaft mbH and concluded on 22 December 1998. The Department CC33 of Industrieanlagen-Betriebsgesellschaft mbH is an evaluation facility recognised by BSI (ITSEF)[7].

The sponsors are Bull S.A. and IBM Informationssysteme Deutschland GmbH. The developers are Bull S.A. and IBM Informationssysteme Deutschland GmbH. Distributors of the product are Bull S.A. and IBM Informationssysteme Deutschland GmbH.

The certification was concluded with

- the comparability test and
- the preparation of this certification report.

This work was concluded on 11 March 1999.

The confirmation of the evaluation assurance level (EAL) only applies on the condition that

- all stipulations regarding generation, configuration and operation, as given in the following certification results, are observed,
- the product is operated – where indicated – in the environment described.

This certification report is only valid for the version of the product indicated here. The validity of the certificate can be extended to cover new versions and releases of the product, provided the applicant applies for re-certification of the modified product, in accordance with the procedural requirements, and provided the evaluation does not reveal any security deficiencies.

With regard to the meaning of the evaluation assurance levels (EAL) and the strength of function (SoF), please refer to the excerpts from the criteria.

[7] Information Technology Security Evaluation Facility

4 Publication

The following Certification Results consist of pages B-1 to B-24.

The product B1/EST-X Version 2.0.1 with AIX Version 4.3.1 has been included in the BSI list of certified products which is published at regular intervals (e.g. in the Internet at http://www.bsi.bund.de). Informationen can be obtained via the BSI-Infoline +49-228/9582-111.

Further copies of this certification report may ordered from the sponsor[8] of the product. The certification report may also be obtained in electronic form at the internet address stated above.

[8] Bull S.A., 1 Rue de Provence, BP 208, 38432 Echirolles Cedex – France and IBM Informationssysteme Deutschland GmbH, Anzinger Straße 29, 81671 München

B Certification Results

The following results represent a summary of

- the security target of the sponsor for the target of evaluation,
- the corresponding evaluation results of the accredited and licensed evaluation facility,
- supplementary notes and stipulations of the certification body.

Contents of the Certification Results

1 Executive Summary

1.1 Description of the TOE

The TOE, B1/EST-X Version 2.0.1 with AIX Version 4.3.1, is a general purpose multi-user, multi-level operating system fulfilling the requirements of functionality class F-B1 of [ITSEC] conformant with part 2 of CC. The following security functionality is provided by the TOE:

1.1.1 Mandatory Access Control (MAC)

The TOE's solution for mandatory access control allows most users to work on the system without being cognizant of the controls, yet allows the administrator to monitor and protect information at multiple MAC levels. The system supports up to 256 hierarchical classifications and up to 128 non-hierarchical compartments. The TOE labels every object (file, device) with a MAC label and enforces mandatory access control between all subjects (i.e. processes) and objects.

The TOE enforces a hierarchical directory tree with increasing MAC levels of files descending in the hierarchy. Thus, every file and directory dominates the MAC level of its parent directory. The TOE implements multilevel directories for those directories which are shared by convention in traditional UNIX, e.g. /tmp. Each directory is actually a collection of directories, each of which contains all files at a single MAC level. This technique is used to implement all directories which are conventionally shared in UNIX. A user can only see files whose MAC level is the same as the process MAC level. Administrative programs, such as spooling programs, can view files in all directories to collect work from all users on the system. Thus, regular users are given the impression that /tmp contains files at their MAC level only, while administrative users can manipulate the collection of directories which together implement /tmp.

The use of each device is under the control of the administrator. The functionality class F-B1 ([ITSEC]) specifies different treatment for single- and multi-level devices, with different handling requirements for each. For tapes, the administrator controls the maximum MAC level of information imported to, or exported from, that device. For each printer, the administrator controls whether single- or multi-level information can be printed and the maximum level of informationen to be printed. Each device node in the file system is assigned according to the requirements of the Security Officer.

The TOE associates a current MAC level and clearance with each user process. A process may only write to a file that is at the same MAC level as the process. A user may not see the contents of a file that is above his clearance. User clearances are managed by the administrator and are stored together with all the additional per-user and system-wide parameters.

1.1.2 Discretionary Access Control (DAC)

The TOE provides the traditional UNIX features that allow read, write and execute (search) permissions for the owner, the group, and others (OGO

protection). Also provided is a system of discretionary access controls based on access control lists, giving finer granularity of control.

1.1.3 Identification and Authentication (I&A)

The TOE implements the password recommendations of the DoD NCSC Green Book ([GREEN]).

1.1.4 Audit

The TOE includes a highly configurable audit system. The generation of audit records is controlled by system default and per-user audit masks. This granularity allows the administrator to generate only the audit information required at that site, minimizing the size of the audit trails.

The audit system has minimal overhead, as the audit data is buffered within the kernel and sent to an audit daemon outside the kernel. The audit daemon then compacts the data and writes it to a file. The size of the buffers, use of compaction, and other parameters are all configured by editing text files.

In addition to the pre-defined audit events, applications can generate their own audit records and add them to the system audit trail.

1.1.5 Data Interchange (Import/Export)

The import/export features of the TOE allow the system to produce and manipulate magnetic export media. In addition, printed information is labeled so that it can be physically handled in accordance with its MAC label. The TOE recognizes the following tape formats:

- Traditional *tar* and *cpio*
- POSIX-compliant *pax*
- Multilevel *lpax*

The Security Officer designates each tape device as single- or multi-level (unlabeled or labeled) formats. Labeled formats include a tape header which describes the security configuration of the system and parameters for each configured security policy and a body which describes the security parameters of each field preceding the file's contents.

A new utility, *lpax*(1) based on the POSIX portable archiver *pax*, has been implemented to manipulate multi-level media. This assures that the use of each device is consistent with its assignment. *lpax* recognizes ordinary tar, cpio and pax formats and creates multi-level archives based on the pax format.

The backup and restore utilities may also be used for import and export of labeled data, but these are only available to the root user and are only intended for use in system installation and in the manufacture of installable media. Since the root user is allowed to bypass MAC controls, MAC mediation has not been implemented in these utilities.

The import and export channels supported by the TOE are diskette and tape devices.

1.1.6 Object Reuse

The TOE satisfies the ITSEC F/B1 requirements for object reuse. The object reuse mechanism is totally transparent and is invoked automatically. It cannot be disabled, even by an administrator.

1.2 Assurance Package

The TOE meets the assurance requirements of assurance level EAL 4 augmented by part 3 assurance component ALC_FLR.2 (Life Cycle Support – Flaw Reporting Procedures).

1.3 Functionality

The TOE security functions fulfill the requirements of functionality class F-B1 ([ITSEC]). All requirements are mapped to functional requirements of CC part 2, this means the TOE is part 2 conformant. The relevant part 2 functional components and the specifications of the TOE security functions are provided in the security target which is attached as part D of this certification report.

1.4 Strength of Functions

The TOE‘s strength of functions is rated ‘high’ (SOF-high).

1.5 Summary of threats and Organisational Security Policies (OSPs) addresssed by the evaluated IT product

The TOE is able to counter threats which may be broadly categorized as the threat of attack from hostile outsiders with no legitimate access to the system and threats from insiders with legitimate access to the system attempting to gain access to and perform operations on objects for which they have no individually defined rights. In addition, certain threats of a non-IT nature can affect the security of the TOE and must be dealt with by the operating environment. The threats which were assumed for the evaluation are specified in the security target which is attached as part D of this certification report.

1.6 Special configuration requirements

Details on secure TOE configuration can be found in the document "Software Release Bulletin" [SRB431] which is delivered with the TOE.

1.7 Assumptions about the operating environment

The TOE is running on the following hardware platforms:

- IBM S-70 (RS64 processors),
- Bull Escala RL 470 (RS64 processors, OEM version of S-70),
- Bull Escala T Series (with 2 up to 4 PowerPC604),
- IBM F-50 (with 2 up to 4 PowerPC604).

The following peripherals can be run with the TOE preserving the security functionality:

- Qume QVT61 terminals operating in VT220 mode only, or terminals which emulate QVT61 VT220 mode completely.
- all storage devices and backup devices supported by the TOE (hard disks, CD-ROM drives,streamer drives, floppy disk drives)

Further assumptions on secure usage are detailed in the security target which is attached as part D of this certification report.

1.8 Disclaimers

The Certification Results only apply to the version of the product indicated in the Certificate and on the condition that all the stipulations are kept with regard to generation, configuration and operation as detailed in this Certification Report.

This certificate is not an endorsement of the IT product by the Bundesamt für Sicherheit in der Informationstechnik or any other organisation that recognises or gives effect to this certificate, and no warranty of the IT product by Bundesamt für Sicherheit in der Informationstechnik or any other organisation that recognises or gives effect to this certificate, is either expressed or implied.

2 Identification

The Target of Evaluation consists of software and documentation, summarized in the following table:

No	Type	Identifier	Form of Delivery
1	SW	Bull B1/EST-X system, Version 2.0.1	CD-ROM
2	DOC	[SRB431] Software Release Bulletin for 2.0.1, October 5, 1998	Paper
3	SW	AIX, Version 4.3.1	Streamer-Tape, CD-ROM
4	DOC	[AIXONL] Online documentation for AIX 4.3.1 (installed with AIX)	Streamer-Tape, CD-ROM
5	DOC	[BX86A261AQ] B1/EST-X/B1 Secure Features Users Guide, 86 A2 61AQ, Rev. 03 - Official release	Paper
6	DOC	[BX86A262AQ] B1/EST-X/B1 Trusted Facilities Manual, 86 A2 62AQ, Rev. 03 - Official release	Paper
7	DOC	[BX86A264AQ] B1/EST-X/B1 Reference Manual (Secure API), 86 A2 64AQ, Rev. 04 - Official release	Paper

Deliverables of the TOE

On AIX 4.3.1 the TCB option needs to be activated and the Program Temporary Fixes (PTFs) U455989.ptf, U456045.ptf and U456165.ptf need to be applied to the system.

3 Security Policy

Due to the F-B1 functionality of the TOE the underlying model is the information flow model of Bell-LaPadula. Bell-LaPadula covers the mandatory access policy. The model implemented by the TOE is a little more restricted in allowing writing only at the same level (no writing up).

The access rules for plain Bell-LaPadula are defined as follows:

- Write Rule: Subjects can only write to objects with the same or higher MAC label.
- Read Rule: Subjects can only read from objects with the same or lower MAC label.
- Order Rules:

  A MAC label consists of a classification part consisting of a classification level and a category part consisting of a set of categories. Classifications are hierarchical, categories non-hierarchical.

  MAC label A is equal to MAC label B if and only if the classification levels and the category sets are equal.

  MAC label A dominates MAC label B if and only if the classification level of A dominates the classification level of B and the category set of A is a superset of the category set of B.

  If MAC label A and B are neither equal nor one dominating the other, they are incomparable.

The model for the TOE enhances this model in the following way:

- Subject Creation Rule: Subjects are created with the same MAC label as the parent subject with the exceptions of Login (tsm) and explicitly creating a shell at a different level (shmac)
- Object Creation Rule: Objects are created with the same MAC label as the creating subject
- Directory model, non decreasing rule: Files within directories can be only at the same MAC level as the containing directory, directories at the same or higher MAC level. This rule is always enforced, i.e. also for the administrator having MacBypass authorization.
- Write Rule: Subjects can only write to objects with the same MAC label.
- Subject Information Flow Rule: Subjects can only obtain information on the status of other subjects (i.e. processes) which are on the same or lower MAC level.
- Bypass Rule: Only users with MacBypass authorization are allowed to bypass MAC mediation (with the exception of a violation of the directory model, see above)

Given these rules, the following two properties are valid:

- Simple Security Property:

  For a subject S having read access to object O the MAC label of S must dominate the MAC label of O.

- Star Property (modified):

  For a subject S having write access to object O the MAC label of S must be equal to the MAC label of O.

4 Assumptions and Clarification of Scope

4.1 Usage Assumptions

The TOE is assured to provide effective security measures only if it is installed, managed and used correctly. The TOE must be managed according to the requirements set forth in the TOE's documentation for delivery, operation and user and administrator guidance.

The specific conditions which are assumed to exist are specified in chapter 3 of the security target which is attached as part D of this certification report.

4.2 Environmental Assumptions

The operational environment must be managed according to the requirements set forth in the TOE's documentation for delivery, operation and user and administrator guidance.

The specific conditions which are assumed to exist are specified in chapter 3 of the security target which is attached as part D of this certification report.

4.3 Clarification of Scope

The assumed threats discussed below must be countered in order to support the TOE security capabilities but are not addressed directly by the TOE itself. Such threats must be addressed by the operating environment.

T.INSTALL The TOE may be delivered and installed in a manner which undermines security.

The security offered by TOE is predicated upon the TOE being initially established in a secure state. That includes assurance that the TOE delivered is that which was evaluated and that the TOE is subsequently installed properly.

T.OPERATE Security failures may occur because of improper administration and operation of the TOE.

The security offered by the TOE can be assured only to the extent that the TOE is operated correctly by system administrators and authorized users.
Users or external threat agents may, through accidental discovery or directed search, discover inadequacies in the security

administration of the TOE which permit them to gain logical access to its resources in breach of any permissions they may have.
Potential attackers may seek to develop methods whereby the improperly administered security functions of the TOE may be circumvented during normal operation.

T.PHYSICAL Security-critical parts of the TOE may be subjected to physical attack which may compromise security.

The security offered by the TOE can be assured only against attacks on the TOE which seek to exploit its legitimate interfaces. It is therefore assumed that adequate physical controls are in place to prevent potential attack agents from gaining access to the TOE or the platform upon which the TOE is operating.

5 Architectural Information

5.1 Definitions and Requirements

The TSF of any secure system is the central core which implements all security enforcing functions. The TSF must be protected against corruption, subversion or deactivation, and it must not be possible to bypass its controls. Inside the TOE, this protection is provided directly by hardware as well as by software built on top of the hardware.

5.2 Hardware Support

Basis of all platforms supporting the TOE are the Power and PowerPC architecture. There are two features of these architectures that provide the hardware protection required by the TOE: a conventional paged memory management unit (PMMU) and a two-level privilege model.

The PMMU allows any process, within the TSF or outside, to be given controlled access to a well defined set of memory pages and no others. This allows all processes to be kept separate except where interaction is expressly desired.

The two-level privilege model provides for a supervisor level and a user level. While running at supervisor level, all on-chip resources are accessible, including the registers controlling the PMMU, thus making all memory accessible. When running at user level, the machine state register and the PMMU control registers as well as all other registers and instructions for controlling the accessibility of memory and other resources are inaccessible, so providing a controlled and self-contained domain of execution. The only way of switching into supervisor level is by means of an exception, either generated externally (for example by a peripheral needing service) or generated internally (for example a system call, a page fault or a protection violation).

5.3 Kernel Protection

The AIX kernel runs at supervisor level. It uses the two features described above to ensure that its own resources cannot be compromised or corrupted by

any user program. Whenever it is entered (by an exception) it performs whatever action is required in a secure manner. This may involve manipulating the resources under its sole control on behalf of the caller, for example manipulating the page tables to give the caller more memory. By manipulating shared resources on behalf of the caller, the kernel is able to maintain system security and integrity.

5.4 User-level Protection

5.4.1 Kernel Privilege

Although it would be possible to implement all security enforcing functions within the kernel, the increased size and complexity would make it harder to assure its correctness. Therefore the kernel provides facilities to allow some of its privileges to be conferred upon privileged user programs. This is done by recognizing the root user (user id zero). For root, discretionary and mandatory access controls are not enforced, and certain privileged operations appropriate only to administrators are allowed.

There are three ways in which a process can acquire user id root and hence the root privileges:

- the process may be a descendant of a tsm process (controlling login) which identified and authenticated root as the logged in user
- the process may be a descendant of the init process (the ultimate ancestor of all processes, which runs under user id root) but not of tsm
- the program may be setuid to root, or a descendant of setuid-root program.

The last of these is a facility allowing root privileges to be extended, in a strictly controlled way, to normal users. A program which is setuid to root (or some other user id) confers the owner's effective user id upon the caller for the duration of execution of the program only. The program is then free to perform privileged operations on behalf of the caller. If required, access to the program may be restricted to any set of users by applying discretionary access controls to it.

5.4.2 Trusted Subsystems

The discretionary access controls allow suites of user programs to set up their own privileged domains without the use of any privilege derived from the kernel. Such a suite is known as a „trusted subsystem". The suite is set up to run under its own user or group id and its resources are set to be accessible only to the user or members of the group. Access to the resources of the suite is provided in a controlled way by programs which are setuid or setgid; this means that for the duration of their execution only, the caller of such a program is given the effective identity of the subsystem, so allowing the program access its resources on the caller's behalf. An example of the use of this mechanism is the chsh command which is setgid to group security. This gives it access to appropriate files enabling it to change a user's shell securely. If a trusted subsystem is set up with userid 0 (superuser) commands running within it will

have the special privileges associated with the superuser, and can offer services implemented by using those privileges to normal users in a controlled way.

The following list gives all the administrative groups on the system as delivered. Only some of them are clearly associated with an administrative function, and hence with associated setgid commands. Others exist because all files must have an owning group, and beside of that for historical reasons. Unprivileged users would never be made members of a group associated with an administrative function.

adm	An owning group for certain administrative and accounting files.
audit	Used by the audit trusted subsystem to provide restricted access to the audit facilities.
backup	Used by the backup trusted subsystem for providing restricted access to backup devices.
bin	An owning group for most executables.
cron	Used by the cron trusted subsystem to allow timed and background execution of jobs.
ecs	This group is **not used** in the evaluated configuration.
mail	An owning group for files associated with mail management.
nobody	A default group used by some comms facilities.
printq	Used by the print subsystem to allow it to provide a protected print spooling facility.
security	Used by the security subsystem to provide controlled management of the security database.
staff	A default group for unprivileged users.
sys	An owning group for many system files.
system	Used by certain administrator commands that could be made available to an administrator other than root. However, no such partitioning of roles is claimed.
usr	This group is **not used** in the evaluated configuration.
uucp	An owning group for system files to do with the basic networking option.
perf	This group is **not used** in the evaluated configuration

5.5 Process Separation

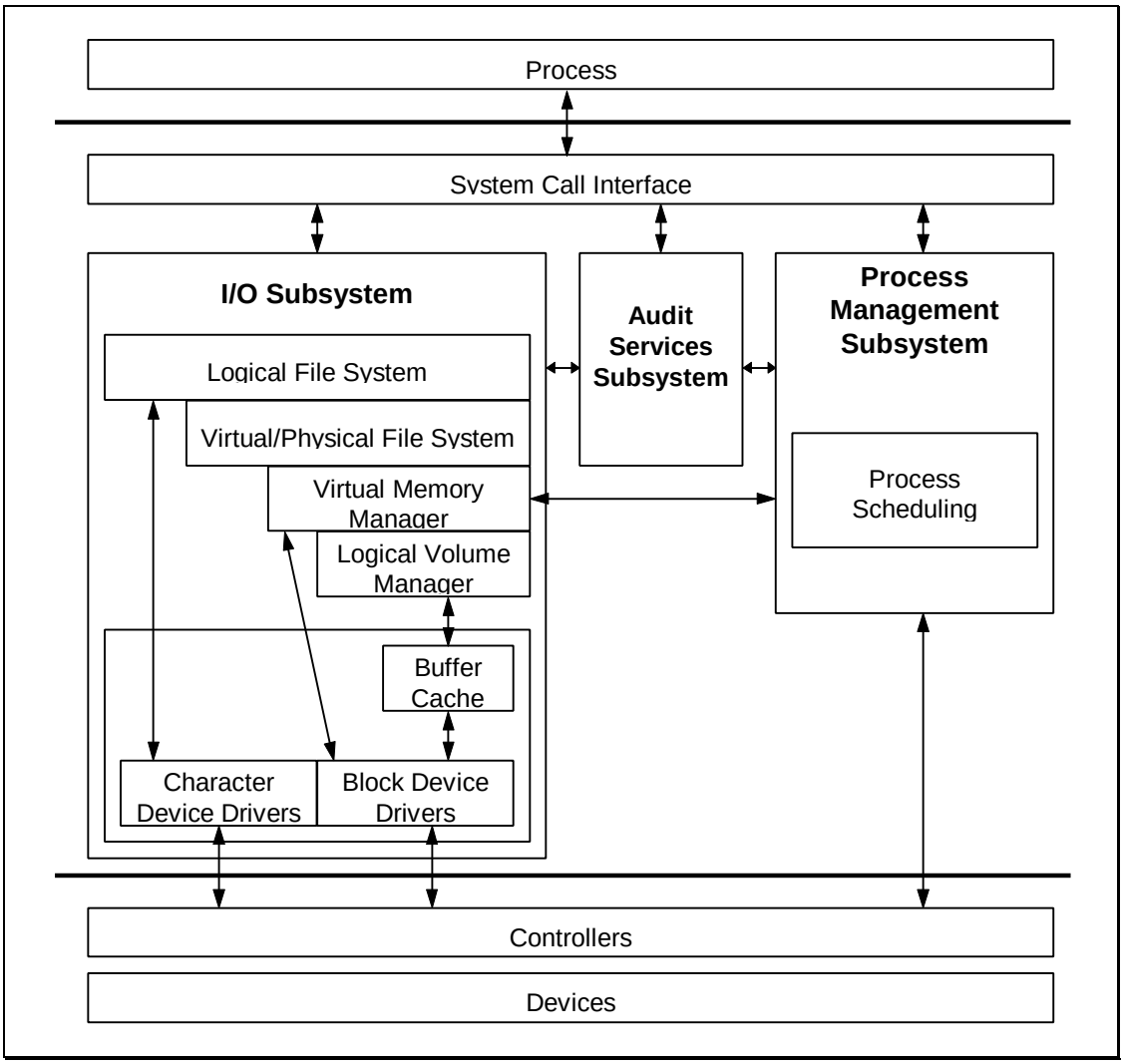
Process separation is achieved with the hardware support described in section 5.2. At user level, each process is given its own exclusive set of pages, except where explicitly requested and mediated by the shared memory facilities. However, within the kernel, processes share access to common data structures. In single processor architectures, this is not normally a problem since the kernel

is coded to run to completion for any request. However, the AIX kernel is pre-emptive, allowing kernel processing to be pre-empted by higher priority processing which may become runable as a result of an interrupt. A locking strategy serializes flow through critical regions, so making them single threaded and eliminating any process separation problems. This also provides the serialization required for the Escala multiprocessor architecture, provided only that the locks are multiprocessor-safe. (The latter point is discussed in section 5.11)

5.6 Separation of TSF subsystems

For those security enforcing functions implemented outside the kernel, separation from other code is achieved by process separation, as described above. This protects the integrity of any command, ensuring that no other command can interfere with its operation. Security enforcing commands are grouped together into trusted subsystems, described in section 5.4. This allows them to operate on a common database protected by Unix file access from interference by commands outside the trusted subsystem. This protection of both security enforcing commands and their data ensures that commands which are not security enforcing are security irrelevant.

Many security enforcing functions are implemented within the kernel, and so they do not have the same level of protection from one another or from security-irrelevant kernel code. This is due to the fact that the kernel as a whole runs at the privileged processor level within one address space. Nevertheless, the kernel is organized as a set of well-defined subsystems with limited interaction. Each of these subsystems provides specific functionality that is logically separated from the other subsystems. Examination of the interactions between kernel subsystems reveals that they are limited to the amount necessary to implement the desired functionality. Especially, no evidence of interactions with negative impact on system security can be identified.



Kernel Architecture

5.7 Binary Compatibility

At user level, all hardware models subject to evaluation are binary compatible, such that any user program can be compiled on any model and run on any other. Applications compiled for 64-bit execution environment are not supported by the TOE and are explicitly excluded for the evaluation. All 32-bit C source code is compiled into an instruction set which is executable by all models. Only in the case of some assembler modules are there different version (denoted file.604 and file.p64) for the different platforms. Both versions are provided on the installation media, but only the appropriate one is installed and loaded for execution. This process is completely automatic and invisible. Those functions which are different for the different platforms are at the lowest level in the hardware interface layer of the kernel, well below the level at which any security enforcing functions are implemented.

Since the kernel is pre-emptive, the only logical difference between single and multiprocessor operation is that in the latter, the versions of the locks used assure mutual exclusion between processors.

5.8 External Interfaces

All security enforcing functions provided by the TOE are accessed in one of two ways, depending upon whether they are implemented in the kernel.

All kernel security enforcing functions are accessed through systems calls. A system call is a programmatic interface which causes an exception, and so switches the processor into supervisor state under which the security enforcing function is executed. At the end of the system call, the processor is switched back to user state before control is returned to the caller.

Other security enforcing functions are implemented within setuid or setgid commands. For setuid commands are afforded certain privileges as described above in order to perform their required actions. These privileges lapse completely as the program terminates. A setgid command has privileges, as described above, with respect to a certain subsystem defined in terms of a group id. Again, that privilege dies with the program.

5.9 Definition of the TSF

We are now in a position to define the TSF, i.e. that part of the system within which all security enforcing functions are implemented, and including all code which would be in a position to interfere with the correct operation of security enforcing functions. The TSF comprises:

- the kernel
- all user programs which run under user id root by virtue of being initiated directly by init
- all user-level programs which run setuid to root
- all programs which are part of the TOE which run setgid to any of the administrative groups

The TOE also provides interfaces in terms of system libraries. None of these are security-enforcing in themselves, in as much as no unprivileged user can call upon any of them to perform a security enforcing function. However, some of them are linked with setuid or setgid programs, and within those programs only, such libraries may be in a position to subvert system security. Therefore these libraries do not constitute a part of the TSF in themselves.

5.10 Kernel Overview

For a general view of the kernel, the reader is referred to books like [MAGIC]. Although it describes System V and not AIX, many of the concepts are common. With that in mind it may be used as background reading or considered authoritative subject to confirmation by browsing the source code.

The following description gives a very high-level view of the structure of the kernel in order to facilitate understanding of other sections of this and other documents.

5.10.1 Processes

A process is represented within the kernel by a proc structure. This holds all information that may be needed, even if the process image is not in store, including copies of the real and effective user ids. Associated with each process there is also a u-block holding information about the process which is only required when that process is in-store. Of particular relevance is the pointer U_cred, to the ucred structure, holding credentials for this process, in particular, user and group ids and supplementary groups. All underlying threads of a process inherit information in the u structure: there is no security relevant information in the specific thread structure.

To avoid modifying the proc structure to hold a MAC label, a new table (proc_mac_table) has been created, containing slots with a one-to-one relationship to the table of proc structures, holding the security information of corresponding processes: this security information contains MAC labels. For performance improvements in terms of memory allocation/deallocation, these MAC labels are pointers to remnant MAC label linked list.

5.10.2 Objects

A file system object is represented by an inode. Associated with this is an ACL which in standard AIX is often null.

Every file system object has its security information represented by a security information definition attached to the extended inode in which is stored the MAC label or the upper and lower MAC labels of a multilevel device. The type definition in this security information determines without ambiguity the type of the referenced object, i.e. multilevel directory type, multilevel device type or regular type: the lower and/or the upper labels in the security information are referenced according to this type.

Inter Process Communication (IPC) objects are represented by entries in an object array. A parallel array is used to hold their MAC labels in the same way as for processes.

5.10.3 Memory Management

Memory is divided into pages which may be swapped in and out independently. For each process is allocated a number of pages and those which are currently in store are mapped into the process' address space by the hardware memory management unit. A reference to an unmapped page causes a page fault, which is trapped by the virtual memory manager subsystem of the kernel. This arranges for the page to be brought into store (possibly swapping out a page belonging to another process in order to make room) and mapped into the process' address space. The process is then resumed and continues without any awareness of the interruption.

Each page has attribute bits associated with it to indicate, for example, whether it is writeable, and each process has a page table associated with it so that the kernel can keep track of its pages. Apart from shared executable pages and

pages belonging to shared memory IPC objects, all processes are allocated disjoint page sets. In this way, process separation is achieved.

5.10.4 Firmware

In the case of the TOE, firmware is used in the power-on self test and the system boot. Whilst system boot could be the target of an attack, the firmware could only be made to compromise security if it were made to modify the executable image of the operating system as it was booted. This would be difficult, requiring an intimate knowledge of unpublished interfaces. It would also require physical access in order to bring the system up in diagnostic mode. Since it is assumed (see Part D 'Security Target') that physical access is not available to an attacker, the conclusion is that for practical purposes no firmware is security relevant.

5.10.5 System Calls and Enablers

The following table lists all system calls and enablers. The list may be regenerated by typing the command:

dump -nv /unix /usr/lib/drivers/kextB1|grep SV|awk '{print $8}'|sort|uniq

This dumps the kernel and B1/EST-X extension name tables and selects those names labelled as system calls, sorting the results. In a small number of cases, the system call name obtained by this method is not the same as the interface to it in libc and hence in the man pages. Of the security enforcing system calls, the following are the only such cases:

- libc function fcntl maps onto system call kfcntl
- libc function stat maps onto system call statx
- libc functions read and write map onto system calls kreadv and kwritev
- all variants of exec map onto system call execve.
- libc functions mount and umount map onto system calls vmount and uvmount.

For those system calls which are regarded as security enforcing, the mechanisms operating within them are listed. No system calls are regarded as security relevant but not security enforcing since system calls do not call one another, so there cannot be a system call B which provides functionality that system call A relies upon to implement any security enforcing functionality.

Some of the system calls listed below are private to a security enforcing subsystem and cannot validly be called by an unprivileged user. These system calls are not regarded as components in their own right, but just as subroutines of another component, which simply need to be implemented within the kernel. The audit system calls listed below but not indicated as security enforcing within the audit mechanism are prime examples. But these and any others will be indicated as „private system calls“ in the appropriate SSDS.

Syscall name	Mechanisms
_exit	
_fp_trapstate_ker	
_load	
_lseek	
_nsleep	
_pause	
_sigaction	
_sigpending	
_sigsuspend	
Absinterval	
Accept	
Access	DAC, MAC
Accessx	
Acct	
Adjtime	
Audit	
Audit_objectevents	
Audit_objectlevels	
Audit_softshutdown	
audit_subjectlevels	
Auditbin	
Auditevents	
Auditlog	AUD
Auditobj	
Auditproc	
Bind	
Bindprocessor	
Brk	OR
Chacl	DAC, MAC
Chdir	DAC,MAC
Chmod	DAC, MAC
Chown	DAC, MAC
Chownx	DAC, MAC
Chpriv	
Chroot	DAC
Close	
cmp_swap	
Connect	
Creat	DAC, MAC, OR
Disclaim	
Execve	I&A, DAC, MAC, OR
Faccessx	
Fchacl	DAC
Fchdir	DAC, MAC
Fchmod	DAC, MAC
Fchown	DAC, MAC
Fchownx	DAC, MAC
Fchpriv	
Fclear	
Fork	MAC, OR
fp_cpusync	
Frevoke	I&A
Fscntl	
Fstatacl	DAC, MAC
Fstatfs	
Fstatpriv	
Fstatx	
Fsync	
Ftruncate	

Syscall name	Mechanisms
Getargs	
Getdirent	
Getdomainname	
Getevars	
Getgidx	
Getgroups	
Gethostid	
Gethostname	
Getinterval	
Getkerninfo	
Getpeername	
Getpgrp	
Getpid	
Getppid	
Getpri	
Getpriority	
Getpriv	
Getprocs	
Getrlimit	
Getrusage	
Getsockname	
Getsockopt	
Getthrds	
Gettimer	
Gettimerid	
Getuidx	
Incinterval	
k_getfile_secinfo	MAC
k_mac_get_ipc	MAC
k_mac_get_proc	MAC
k_setfile_secinfo	MAC
k_mac_set_proc	MAC
Kfcntl	MAC
Kfork	
Kgetpgrp	
Kgetsid	
Kill	
Kioctl	
Knlist	
Kreadv	I&A, OR
Kwaitpid	
Kwritev	I&A, OR
Lchown	DAC
Link	DAC, MAC
Listen	
Loadbind	
Loadquery	
Lockf	
Lseek	
k_mac_dominate	MAC
k_mac_equal	MAC
Madvise	
Mincore	
Mkdir	DAC, MAC, OR
Mknod	DAC, MAC, OR
Mlifdir	MAC
Mlifisdir	MAC
Mmap	
Mntctl	

Syscall name	Mechanisms
Mprotect	
msem_remove	
Msgctl	DAC, MAC
Msgget	MAC
Msgrcv	DAC, MAC
Msgsnd	DAC, MAC
Msgxrcv	DAC, MAC
Msleep	
Msync	
Munmap	
Mwakeup	
Mycpu	
Naccept	
Ngetpeername	
Ngetsockname	
Nrecvfrom	
Nrecvmsg	
Nsendmsg	
Nsleep	
Open	DAC, MAC, OR
Openx	DAC, MAC, OR
Pause	
Pipe	
Plock	
Poll	
Privcheck	
Probe	
Profil	
Psdanger	
Ptrace	
Quotactl	
Readlink	DAC, MAC
Reboot	
Recv	
Recvfrom	
Recvmsg	
Reltimerid	
Rename	DAC, MAC
Resabs	
Resinc	
Restimer	
Revoke	
Rmdir	DAC, MAC
Sbrk	OR
Select	MAC
Semctl	DAC, MAC
Semget	MAC
Semop	
Send	
Sendmsg	
Sendto	
Setdomainname	
Seteuid	
Setgid	
Setgidx	
Setgroups	
Sethostid	
Sethostname	
Setpgid	

Syscall name	Mechanisms
Setpgrp	
Setpri	
Setpriority	
Setpriv	
Setreuid	
Setrlimit	
Setsid	
Setsockopt	
Settimer	
Setuid	
Setuidx	I&A
Shmat	DAC, MAC
Shmctl	DAC, MAC
Shmdt	
Shmget	MAC
Shutdown	
Sigaction	
Sigcleanup	
Siglocalmask	
Sigpending	
Sigprocmask	
Sigreturn	
Sigstack	
Sigsuspend	
Socket	
Socketpair	
Statacl	DAC, MAC
Statdevlvl	
Statfs	
Statpriv	
Statx	DAC, MAC
Swapoff	
Swapon	
Swapqry	
Symlink	DAC, MAC
Sync	
Sysconfig	
thread_create	
thread_kill	
thread_self	
thread_setsched	
thread_setstate	
thread_terminate	
thread_terminate_ack	
thread_tsleep	
thread_twakeup	
Times	
Trcgen	
Trcgent	
Trchk	
Trchkg	
Trchkgt	
Trchkl	
Trchklt	
Trchkt	
Trchook	
Truncate	
Ulimit	
Umask	

Syscall name	Mechanisms
Umount	
Uname	
Unameu	
Unamex	
Unlink	DAC, MAC
Unload	
Upfget	

Syscall name	Mechanisms
Upfput	
Usrinfo	
Ustat	
Utimes	DAC, MAC
Uvmount	
Vmount	DAC, MAC
Yield	

5.11 Multiprocessor Operation

The target platforms include symmetrical multiprocessor systems. In such cases, one processor is nominated during system boot to load the system and to perform the initialization of the kernel, but this processor then enables all others, and from then on no individual processor has any privilege or responsibilities above its peers.

Within the kernel, a system of locks is used to serialize critical regions. Different locks are used for protecting different classes of data structure so as to minimize lock contention. For data only accessed by system calls the „simple_lock“ operation is performed, which checks whether the lock is free and sleeps if not. This arrangement is needed to assure the proper operation in the single processor case since the kernel is pre-emptive. In the multiprocessor case it is only necessary to make the locks effective between processors as well as between processes on one processor.

In single processor systems, mutual exclusion between device drivers and interrupt code is traditionally provided by disabling interrupts. In a multi-processor system this is no longer adequate since it does not provide exclusion between processors. The solution is a „disable_lock“ function, which firstly disables interrupts so ensuring mutual exclusion on the current processor. Then it engages a specified lock, looping if necessary, until it is free. A complementary „unlock_enable“ operation is also provided.

In order to make locks effective between processors, a more primitive lock is required to protect the processing of the engagement and disengagement of the lock. In CISC architectures, this is normally provided by a bus-atomic test-and-set instruction which is executed repeatedly until the result indicates that the current processor tested the lock byte zero and set it non-zero. In a RISC architecture (as exemplified by the TOE) this is no longer appropriate because the delay in waiting for the bus would stall the instruction pipeline. Instead, a Load Word and Reserve Index (lwarx) instruction loads from store and places a „reservation“ on that store location, in the process, canceling any other reservation made by another processor. The loaded value is modified and written back with a „Store Word Conditional Indexed“ (stwcx) instruction, which stores the data conditional on the reservation still existing, and indicates success via the Condition Register. On failure, the whole process would typically be repeated until successful. In this way, functionality equivalent to an atomic test-and-set instruction may be achieved.

Memory management is protected by the simple_lock operation to ensure the integrity of the allocation and de-allocation of page table entries and other data structures. In this way, the fact that there may be more than one processor may be ignored.

There are many places in the kernel where a store location must be updated atomically, for example to add a bit to a bit list. A suite of functions is provided for such operations, such as „fetch_and_add“, „fetch_and_or“ and „fetch_and_and“. Two versions of these exist: a version for the Power architecture (single processor models) which disables interrupts to ensure integrity of the operation, and versions for Escala target platforms (PowerPC multiprocessor models) which use the lwarx/stwcx instructions to ensure integrity in a multiprocessor environment.

Outside the kernel, synchronization between threads in multithreaded applications are necessary.

Certain target platforms may be run in single-processor mode, either by enabling only one of multiple processors, or by running it with only one processor physically installed. In this configuration the code used is identical to the multiprocessor case. However, on initialization, the number of processors is recorded and if this is only one, slightly different code paths are taken at a few points, mainly in interrupt handling, for the purposes of optimization. To describe this area would require a detailed description of the hardware architecture as a prerequisite and so is not appropriate in the present context particularly as this is in the hardware abstraction layer of the kernel, well below any level which is cognizant of security concepts. Consequently, the cases where paths differ are listed below with just a simple description of the differing action. Excluding cases relating only to debugging or inapplicable processor types, they are as follows. (All file paths are relative to src/bos/kernel.)

File	Function	Single processor action
proc/init_lock.c	init_locks()	Limit spin locks to single try
si/main.c	main()	Don’t boot other processors or initialize mp lock
ios/intr_init.c	intr_init()	Handling of off-level interrupts
ios/intr.c	i_poll()	Poll lock not needed
ios/intr.c	i_init(), i_clear()	No need to flag whether an interrupt handler is mp-safe.

The above discussion has shown how multiprocessor execution is achieved and that it might be a possible source of system errors in general. But examination during evaluation by source code inspection, testing and vulnerability assessment did not reveal any errors caused by multiprocessor execution.

6 Documentation

Reference	Document
[AIXONL]	Online documentation for AIX 4.3.1 (installed with AIX)
[SRB431]	Software Release Bulletin for 2.0.1, October 5, 1998
[BX86A261AQ]	B1/EST-X/B1 Secure Features Users Guide, 86 A2 61AQ, Rev. 03 - Official release
[BX86A262AQ]	B1/EST-X/B1 Trusted Facilities Manual, 86 A2 62AQ, Rev. 03 - Official release
[BX86A264AQ]	B1/EST-X/B1 Reference Manual (Secure API), 86 A2 64AQ, Rev. 04 - Official release
	Certification Report No 97/81 issued by the Certification Body of the UK IT Security Evaluation and Certification Scheme for the product Bull BEST-X/B1, Version 1.1.1.9
	Certification Report BSI-ITSEC-0138-1998 issued by BSI for the product AIX Version 4.3

7 IT Product Testing

The test results are based based on Bull documentation on testing of the TOE and additional verification by the evaluators during their visit at the developer site in Echirolles, where the evaluators rerun the whole Bull security test suite.

The test procedures themselves did not change since the initial evaluation. Changes to the test suite have been made to include tests that verify the correct implementation of new TOE features e.g. administrative roles and command authorizations.

Independent informal manual tests were made with the test system supplied to the evaluators during the whole period of evaluation.

The testing of the TOE was performed by the developer and the ITSEF on the following workstations:

- IBM Risc System/6000 F-50 dual-processor (PowerPC604)
- Bull Escala T quadri-processor (PowerPC604)
- F50 quadri-processor (PowerPC604)
- S70 quadri-processor (RS64)

8 Evaluated Configuration

Re-Evaluation was performed by the ITSEF on the following configuration:

- System:

  - IBM Risc System/6000 F-50
  - Processor: 2 Power PC604
  - 256 MB RAM
- Peripherals:
  - Two 1,1 GB Harddisks
  - SCSI CD-ROM Drive
  - SCSI Exabyte Drive
  - SCSI DAT Drive
  - 3,5"-Disk Drive
  - Ethernet Card

9 Results of the Evaluation

- The TOE meets the assurance requirements of assurance level EAL4 augmented with ALC_FLR.2 (Life Cycle Support – Flaw Reporting Procedures) according to the CC.
- The TOE's strength of function is rated ‘high’ (SOF-high).

10 Evaluator Comments/Recommendations

Advice and information for the user for the secure operation of the TOE are provided in the document [SRB431], which is delivered with the TOE.

The following advice has to be taken into account by the users:

- The user has to be aware of the known limitations and defects as well as usage considerations of the TOE (see chapter 3.3 of [SRB431])
- Configuration of AIX 4.3.1 must be done in a secure manner following the guidance in chapter 4.2.1 of [SRB431]
- Configuration of B1/EST-X must be done in a secure manner following the guidance in chapter 4.2.2 of [SRB431]
- Applications compiled for 64-bit execution environment are not supported by B1/EST-X and are explicitly excluded from the evaluation
- I&A-extensions: It is possible to implement customer encryption and customer password generation. This requires specific code written by the customer that needs to be integrated into the TOE. No such code additions have been evaluated for obvious reasons. Any implementation of these features integrated by the customer modifies the certified TOE so that the certificate no longer is valid

11 Security Target

The security target for B1/EST-X Version 2.0.1 with AIX Version 4.3.1 is included in part D of this certification report.

12 Glossary

12.1 Acronyms

CC Common Criteria for IT Security Evaluation

EAL Evaluation Assurance Level

IT Information Technology

PP Protection Profile

SF Security Function

SFP Security Function Policy

SOF Strength of Function

ST Security Target

TOE Target of Evaluation

TSC TSF Scope of Control

TSF TOE Security Functions

TSP TOE Security Policy

12.2 Glossary

Augmentation - The addition of one or more assurance component(s) from Part3 to an EAL or assurance package.

Extension - The addition to an ST or PP of functional requirements not contained in Part2 and/or assurance requirements not contained in Part 3 of the CC.

Formal - Expressed in a restricted syntax language with defined semantics based on well-established mathematical concepts.

Informal - Expressed in natural language.

Object - An entity within the TSC that contains or receives information and upon which subjects perform operations.

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

Security Function - A part or parts of the TOE that have to be relied upon for enforcing a closely related subset of the rules from the TSP.

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

Semiformal - Expressed in a restricted syntax language with defined semantics.

Strength of Function - A qualification of a TOE security function expressing the minimum efforts assumed necessary to defeat its expected security behaviour by directly attacking its underlying security mechanisms.

SOF-basic - A level of the TOE strength of function where analysis shows that the function provides adequate protection against casual breach of TOE security by attackers possessing a low attack potential.

SOF-medium - A level of the TOE strength of function where analysis shows that the function provides adequate protection against straightforward or intentional breach of TOE security by attackers possessing a moderate attack potential.

SOF-high - A level of the TOE strength of function where analysis shows that the function provides adequate protection against deliberately planned or organised breach of TOE security by attackers possessing a high attack potential.

Subject - An entity within the TSC that causes operations to be performed.

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

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

TOE Security Policy - A set of rules that regulate how assets are managed, protected and distributed within a TOE.

TSF Scope of Control - The set of interactions that can occur with or within a TOE and are subject to the rules of the TSP.

13 Bibliography

[CC] Common Criteria for IT Security Evaluation, Version 2.0

[CEM] Draft of the Common Methodology for IT Security Evaluation, Version 0.4

[GREEN] DoD Password Management Guideline, CSC-STD-002-85, 12 April 1985

[ITSEC] Information Technology Security Evaluation Criteria (ITSEC) Provisional Harmonised Criteria of France, Germany, the Netherlands and the United Kingdom Version 1.2, June 1991

[ITSEM] Information Technology Security Evaluation Manual (ITSEM) Provisional Harmonised Methodology Version 1.0, September 1993

[MAGIC] The Magic Garden Explained, Berny Goodheart & James Cox, ISBN 0-13-098138-9

C Excerpts from the Criteria

CC Part 1

Caveats on evaluation results (chapter 5.4)

The pass result of evaluation shall be a statement that describes the extent to which the PP or TOE can be trusted to conform to the requirements. The results shall be caveated with respect to Part 2 (functional requirements), Part 3 (assurance requirements) or directly to a PP, as listed below.

a) **Part 2 conformant** - A PP or TOE is Part 2 conformant if the functional requirements are only based upon functional components in Part 2.

b) **Part 2 extended** - A PP or TOE is Part 2 extended if the functional requirements include functional components not in Part 2.

c) **Part 3 conformant** - A PP or TOE is Part 3 conformant if the assurance requirements are in the form of an **EAL** or **assurance package** that is based only upon assurance components in Part 3.

d) **Part 3 augmented** - A PP or TOE is Part 3 augmented if the assurance requirements are in the form of an **EAL** or **assurance package**, plus other assurance components in Part 3.

e) **Part 3 extended** - A PP or TOE is Part 3 extended if the assurance requirements are in the form of an **EAL** associated with additional assurance requirements not in Part 3 or an **assurance package** that includes (or is entirely made up from) assurance requirements not in Part 3.

f) **Conformant to PP** - A TOE is conformant to a PP only if it is compliant with all parts of the PP.

CC Part 3

Assurance categorisation (chapter 2.5)

The assurance classes, families, and the abbreviation for each family are shown in Table 2.1.

Assurance Class	Assurance Family	Abbreviated Name
Class ACM: Configuration management	CM automation	ACM_AUT
	CM capabilities	ACM_CAP
	CM scope	ACM_SCP
Class ADO: Delivery and operation	Delivery	ADO_DEL
	Installation, generation and start-up	ADO_IGS
Class ADV: Development	Functional specification	ADV_FSP
	High-level design	ADV_HLD
	Implementation representation	ADV_IMP
	TSF internals	ADV_INT
	Low-level design	ADV_LLD
	Representation correspondence	ADV_RCR
	Security policy modeling	ADV_SPM
Class AGD: Guidance documents	Administrator guidance	AGD_ADM
	User guidance	AGD_USR
Class ALC: Life cycle support	Development security	ALC_DVS
	Flaw remediation	ALC_FLR
	Life cycle definition	ALC_LCD
	Tools and techniques	ALC_TAT
Class ATE: Tests	Coverage	ATE_COV
	Depth	ATE_DPT
	Functional tests	ATE_FUN
	Independent testing	ATE_IND
Class AVA: Vulnerability assessment	Covert channel analysis	AVA_CCA
	Misuse	AVA_MSU
	Strength of TOE security functions	AVA_SOF
	Vulnerability analysis	AVA_VLA

Table 2.1 -Assurance family breakdown and mapping

Evaluation assurance levels (chapter 6)

The Evaluation Assurance Levels (EALs) provide an increasing scale that balances the level of assurance obtained with the cost and feasibility of acquiring that degree of assurance. The CC approach identifies the separate concepts of assurance in a TOE at the end of the evaluation, and of maintenance of that assurance during the operational use of the TOE.

It is important to note that not all families and components from Part 3 are included in the EALs. This is not to say that these do not provide meaningful and desirable assurances. Instead, it is expected that these families and components will be considered for augmentation of an EAL in those PPs and STs for which they provide utility.

Evaluation assurance level (EAL) overview (chapter 6.1)

Table 6.1 represents a summary of the EALs. The columns represent a hierarchically ordered set of EALs, while the rows represent assurance families. Each number in the resulting matrix identifies a specific assurance component where applicable.

As outlined in the next section, seven hierarchically ordered evaluation assurance levels are defined in the CC for the rating of a TOE's assurance. They are hierarchically ordered inasmuch as each EAL represents more assurance than all lower EALs. The increase in assurance from EAL to EAL is accomplished by *substitution* of a hierarchically higher assurance component from the same assurance family (i.e. increasing rigour, scope, and/or depth) and from the *addition* of assurance components from other assurance families (i.e. adding new requirements).

These EALs consist of an appropriate combination of assurance components as described in chapter 2 of this Part 3. More precisely, each EAL includes no more than one component of each assurance family and all assurance dependencies of every component are addressed.

While the EALs are defined in the CC, it is possible to represent other combinations of assurance. Specifically, the notion of “augmentation“ allows the addition of assurance components (from assurance families not already included in the EAL) or the substitution of assurance components (with another hierarchically higher assurance component in the same assurance family) to an EAL. Of the assurance constructs defined in the CC, only EALs may be augmented. The notion of an “EAL minus a constituent assurance component“ is not recognised by the CC as a valid claim. Augmentation carries with it the obligation on the part of the claimant to justify the utility and added value of the added assurance component to the EAL. An EAL may also be extended with explicitly stated assurance requirements.

Assurance Class	**Assurance Family**	**Assurance Components by Evaluation Assurance Level**						
		EAL1	**EAL2**	**EAL3**	**EAL4**	**EAL5**	**EAL6**	**EAL7**
Configuration management	ACM_AUT				**1**	1	**2**	2
	ACM_CAP	**1**	**2**	**3**	**4**	4	**5**	5
	ACM_SCP			**1**	**2**	**3**	3	3
Delivery and operation	ADO_DEL		**1**	1	**2**	2	2	**3**
	ADO_IGS	**1**	1	1	1	1	1	1
Development	ADV_FSP	**1**	1	1	**2**	**3**	3	**4**
	ADV_HLD		**1**	**2**	2	**3**	**4**	**5**
	ADV_IMP				**1**	**2**	**3**	3
	ADV_INT					**1**	**2**	3
	ADV_LLD				**1**	1	**2**	2
	ADV_RCR	**1**	1	1	1	**2**	2	**3**
	ADV_SPM				**1**	**3**	3	3
Guidance documents	AGD_ADM	**1**	1	1	1	1	1	1
	AGD_USR	**1**	1	1	1	1	1	1
Life cycle support	ALC_DVS			**1**	1	1	**2**	2
	ALC_FLR							
	ALC_LCD				**1**	**2**	2	**3**
	ALC_TAT				**1**	**2**	**3**	3
Tests	ATE_COV		**1**	**2**	2	2	**3**	3
	ATE_DPT			**1**	1	**2**	2	**3**
	ATE_FUN		**1**	1	1	1	**2**	2
	ATE_IND	**1**	**2**	2	2	2	2	**3**
Vulnerability assessment	AVA_CCA					**1**	**2**	2
	AVA_MSU			**1**	**2**	2	**3**	3
	AVA_SOF		**1**	1	1	1	1	1
	AVA_VLA		**1**	1	**2**	**3**	**4**	4

Table 6.1 - Evaluation assurance level summary

Evaluation assurance level 1 (EAL1) - functionally tested (chapter 6.2.1)

Objectives

EAL1 is applicable where some confidence in correct operation is required, but the threats to security are not viewed as serious. It will be of value where independent assurance is required to support the contention that due care has been exercised with respect to the protection of personal or similar information.

EAL1 provides an evaluation of the TOE as made available to the customer, including independent testing against a specification, and an examination of the guidance documentation provided. It is intended that an EAL1 evaluation could be successfully conducted without assistance from the developer of the TOE, and for minimal outlay.

An evaluation at this level should provide evidence that the TOE functions in a manner consistent with its documentation, and that it provides useful protection against identified threats.

Evaluation assurance level 2 (EAL2) - structurally tested (chapter 6.2.2)

Objectives

EAL2 requires the co-operation of the developer in terms of the delivery of design information and test results, but should not demand more effort on the part of the developer than is consistent with good commercial practice. As such it should not require a substantially increased investment of cost or time.

EAL2 is therefore applicable in those circumstances where developers or users require a low to moderate level of independently assured security in the absence of ready availability of the complete development record. Such a situation may arise when securing legacy systems, or where access to the developer may be limited.

Evaluation assurance level 3 (EAL3) - methodically tested and checked (chapter 6.2.3)

Objectives

EAL3 permits a conscientious developer to gain maximum assurance from positive security engineering at the design stage without substantial alteration of existing sound development practices.

EAL3 is applicable in those circumstances where developers or users require a moderate level of independently assured security, and require a thorough investigation of the TOE and its development without substantial re-engineering.“

Evaluation assurance level 4 (EAL4) - methodically designed, tested, and reviewed (chapter 6.2.4)

Objectives

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

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

Evaluation assurance level 5 (EAL5) - semiformally designed and tested (chapter 6.2.5)

Objectives

EAL5 permits a developer to gain maximum assurance from security engineering based upon rigorous commercial development practices supported by moderate application of specialist security engineering techniques. Such a TOE will probably be designed and developed with the intent of achieving EAL5 assurance. It is likely that the additional costs attributable to the EAL5 requirements, relative to rigorous development without the application of specialised techniques, will not be large.

EAL5 is therefore applicable in those circumstances where developers or users require a high level of independently assured security in a planned development and require a rigorous development approach without incurring unreasonable costs attributable to specialist security engineering techniques.

Evaluation assurance level 6 (EAL6) - semiformally verified design and tested (chapter 6.2.6)

Objectives

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

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

Evaluation assurance level 7 (EAL7) - formally verified design and tested (chapter 6.2.7)

Objectives

EAL7 is applicable to the development of security TOEs for application in extremely high risk situations and/or where the high value of the assets justifies the higher costs. Practical application of EAL7 is currently limited to TOEs with tightly focused security functionality that is amenable to extensive formal analysis.

Strength of TOE security functions (AVA_SOF) (chapter 14.3)

AVA_SOF Strength of TOE security functions

Objectives

Even if a TOE security function cannot be bypassed, deactivated, or corrupted, it may still be possible to defeat it because there is a vulnerability in the concept of its underlying security mechanisms. For those functions a qualification of their security behaviour can be made using the results of a quantitative or statistical analysis of the security behaviour of these mechanisms and the effort required to overcome them. The qualification is made in the form of a strength of TOE security function claim.

Vulnerability analysis (AVA_VLA) (chapter 14.4)

AVA_VLA Vulnerability analysis

Objectives

Vulnerability analysis is an assessment to determine whether vulnerabilities identified, during the evaluation of the construction and anticipated operation of the TOE or by other methods (e.g. by flaw hypotheses), could allow users to violate the TSP.

Vulnerability analysis deals with the threats that a user will be able to discover flaws that will allow unauthorised access to resources (e.g. data), allow the ability to interfere with or alter the TSF, or interfere with the authorised capabilities of other users.

Application notes

A vulnerability analysis is performed by the developer in order to ascertain the presence of security vulnerabilities, and should consider at least the contents of all the TOE deliverables including the ST for the targeted evaluation assurance level. The developer is required to document the disposition of identified vulnerabilities to allow the evaluator to make use of that information if it is found useful as a support for the evaluator's independent vulnerability analysis.“

„Independent vulnerability analysis goes beyond the vulnerabilities identified by the developer. The main intent of the evaluator analysis is to determine that the TOE is resistant to penetration attacks performed by an attacker possessing a low (for AVA_VLA.2), moderate (for AVA_VLA.3) or high (for AVA_VLA.4) attack potential.

D Security Target