National Information Assurance Partnership

Common Criteria Evaluation and Validation Scheme Validation Report

International Business Machines

Global Security Kit

Version 7.0.4.11

Report Number: CCEVS-VR-07-0039

Dated: 02 August 2007

Version: 1.4

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

Orion Security Solutions
Arlington, VA

atsec Information Security Corporation

Austin, TX

Table of Contents

1. EXECUTIVE SUMMARY 4
2. IDENTIFICATION 4
3. SECURITY POLICY 5
   - 3.1. KEY MANAGEMENT 5
   - 3.2. CRYPTOGRAPHIC ALGORITHMS 6
   - 3.3. SECURE CHANNELS 6
   - 3.4. SELF-TESTS AND FAILURE HANDLING 7
4. ASSUMPTIONS 7
   - 4.1. USAGE ASSUMPTIONS 7
   - 4.2. CLARIFICATION OF SCOPE 8
5. ARCHITECTURAL INFORMATION 9
6. DOCUMENTATION 10
7. IT PRODUCT TESTING 10
   - 7.1. DEVELOPER TESTING 10
   - 7.2. EVALUATOR TESTING 11
8. EVALUATED CONFIGURATION 13
9. RESULTS OF THE EVALUATION 14
10. VALIDATOR COMMENTS 15
11. SECURITY TARGET 15
12. LIST OF ACRYONYMS 17
13. BIBLIOGRAPHY 19

1. EXECUTIVE SUMMARY

This report documents the NIAP validators' assessment of the evaluation of International Business Machines (IBM) Global Security ToolKit (GSKit) Version 7.0.4.11. It presents the evaluation results, their justifications, and the conformance results. This validation report is not an endorsement of the IT product by any agency of the U.S. Government and no warranty of the IT product is either expressed or implied.

The evaluation was performed by the atsec Information Security Corporation, and was completed during June 2007. 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 **EAL4.**

Global Security ToolKit (GSKit) Version 7.0.4.11 is a set of tools and C/C++ programming interfaces that can be integrated in software applications to add secure channels using the SSLv3 and TLSv1 protocols. It provides the cryptographic functions, the protocol implementation, and key generation and management functionality for this purpose. GSKit is a software only component; the platform/hardware is part of the TOE environment.

The main function of GSKit is to provide a secure channel to another trusted IT product. GSKit Version 7.0.4.11 encapsulates the IBM Crypto for C (ICC) Version 1.4.5 component, which provides cryptographic functions. A large subset of the algorithms implemented in the ICC module and used by GSKit Version 7.0.4.11 have been validated according to FIPS 140-2 under the Cryptographic Module Validation Program.

The evaluation covers a wide range of operating systems and associated platforms running the evaluated configurations of the TOE.

The validation team monitored the activities of the evaluation team, provided guidance on technical issues and evaluation processes, reviewed successive versions of the Security Target, reviewed selected evaluation evidence, reviewed test plans, reviewed intermediate evaluation results (i.e., the CEM work units), and reviewed successive versions of the evaluation technical report (ETR) and test report. The validation team determined that the evaluation team showed that the product satisfies all of the functional requirements and assurance requirements defined in the Security Target (ST) an EAL4 evaluation. Therefore, the validation team concludes that the CCTL findings are accurate, and the conclusions justified.

2. IDENTIFICATION

The CCEVS is a joint National Security Agency (NSA) and National Institute of Standards and Technology (NIST) 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.

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

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

- The Target of Evaluation (TOE): the fully qualified identifier of the product as evaluated;
- The Security Target (ST), describing the security features, claims, and assurances of the product;
- The conformance result of the evaluation;
- 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 Global Security ToolKit v7.0.4.11
Protection Profile	None
Security Target	*IBM Global Security Kit Version 7.0.4.11 Security Target;* Version 1.7, 26 July 2007
Evaluation Technical Report	*Evaluation Technical Report for a Target of Evaluation: IBM Global Security Kit (GSKit) Version 7.0.4.11 with IBM Crypto for C (ICC) 1.4.5.* Version 1.3, 02 August 2007
Conformance Result	CC V2.3, Part 2 extended, Part 3 conformant, EAL 4
Sponsor	International Business Machines (IBM)
Developer	International Business Machines (IBM)
Evaluators	atsec Information Security Corporation
Validators	Orion Security Solutions

3. SECURITY POLICY

3.1. Key Management

GSKit creates, imports, and maintains the keys and certificates contained in the keystore database. The KM (Key Management) API provides an extensive list of functions to manage this database. In addition, a command line interface (CLI) allows users direct access to a selected subset of the KM API.

Access to the keystore is password-protected. Private keys that are stored in the keystore are encrypted prior to storage and decrypted after retrieval to ensure their confidentiality. A HMAC-SHA-1 hash is generated over all data stored in the keystore and stored separately in the keystore to ensure integrity of the data. PKCS#11 devices that have been FIPS 140-2 certified can be utilized to retrieve and/or store public key certificates, private keys and security attributes and to request the execution of cryptographic primitives.

Certificates conformant to X.509 are used for SSL/TLS connections. These certificates can be imported and exported using either the Key Management API or the CLI. Certificate requests conformant to PKCS#10 can be generated and used to request the signing of the public key used to create a certificate. The import and export of cryptographic material in PKCS#12 format files is also supported.

3.2. Cryptographic Algorithms

GSKit may contain a variety of software libraries and hardware accelerators to support the cryptographic functions required for the SSL/TLS protocols. The evaluated configuration is limited to the IBM Crypto For C (ICC) library and PKCS#11 devices as cryptographic service providers.

All cryptographic algorithms used within the cipherspec portions of the SSL/TLS ciphersuites are FIPS approved. Other cryptographic algorithms may be used for during the setup of SSL/TLS connections. A FIPS 140-2 approved random number generation function is also available.

3.3. Secure Channels

GSKit offers the possibility to set up secure/trusted channels using the SSLv3 or TLSv1 protocol, earlier versions of SSL are disabled in the evaluated configuration. These channels offer confidentiality and integrity protection of the data transmitted over them. This functionality is accessible through the SSL API of the GSKit. It is both possible to write client- and server-applications using this API and the libraries are capable of multi-threading.

For authentication, X.509 certificates as defined in the X.509 standard are used.

GSKit may operate in server or client mode. When in client mode, server authentication (with a valid X.509 certificate and the associated private key) is required for the successful completion of the SSLv3 and TLSv1 channels. This means that the session is only established if the certificate of the communication partner can be validated by the TOE and the communication partner proves that it has the associated private key.

When in server mode, GSKit can be configured whether client authentication is required. If client authentication is required, then client authentication (with a valid X.509 certificate and the associated private key) is needed for the successful establishment of the SSLv3 and TLSv1 channels without restrictions. If the communication partner presents a certificate that GSKit can not validate or a certificate verified as invalid (due to expiration or due to being revoked), the session is terminated. If client authentication is required and the client does not present any certificate the session will be established and a flag set indicating that the communication partner did not present a certificate. The application using GSKit can check this flag and decide to terminate the session, send an error message to the communication partner and then terminate the session or to continue the session with a set of services adapted to the fact that the communication partner did not authenticate itself. If client authentication is not required, no authentication of the communication partner is performed.

GSKit can be configured to perform certificate validation with either CRLs and/or an OCSP responder. Certificate validation using CRLs and/or OCSP is not mandatory.

Sensitive data is protected between different sessions. All unprotected (i.e. cleartext) critical security parameters and random numbers are considered sensitive data and are cleared before the buffer containing them is released. The random number generator clears it state automatically when the module is removed from memory.

3.4. Self-tests and Failure Handling

Self-test functionality within the ICC component is called automatically with the first call to GSKit for SSL environment initialization. This ensures that self-tests are performed before the first use of the cryptographic algorithms for the setting up of a SSL/TLS connection. The self-tests check cryptographic operations and the software integrity.

Upon detection of an error condition, GSKit will enter a failure mode while preserving a secure state.

4. ASSUMPTIONS

4.1. Usage Assumptions

As stated in the Security Target, the assumptions made regarding the minimum physical and procedural measures to maintain security of the TOE are:

- Those responsible for the administration of the TOE and the TOE environment are competent and trustworthy individuals, capable of managing the TOE and the TOE environment and the security of the information it contains.
- The TOE environment is responsible for the management and generation of the certificates.
- Those who integrate the TOE as part of a larger product or system are assumed to follow the guidance provided with the TOE with respect to the configuration of the TOE, the use of TOE functions and interfaces and the protection of the TOE code and data. They are trustworthy and do not try to subvert or bypass TOE security functions;
- Users knowing the password to protect the keystore will apply high protection measures that prohibit unauthorized access to or disclosure of password information.
- The environment will provide reliable timestamps;
- OCSP responders used by the TOE are trustworthy;
- If a CRL is required by the TOE, the TOE environment must provide a valid, current CRL.

4.2. Clarification of Scope

GSKit is a collection of functions organized as a dynamically linked library (DLL) installed as a separate product that any authorized application program running on the platform may call to establish an SSL/TLS communication link. When called by an application, GSKit inherits the permissions and privileges of the calling program and therefore must rely on the operating system, and application program to provide the essential security features.

GSKit provides a set of tools and C/C++ programming interfaces that can be used to add secure channels using the SSLv3 and TLSv1 protocols to TCP/IP applications (products). It provides the cryptographic functions, the protocol implementation and key generation and management functionality for this purpose.

GSKit ships only compiled object code and header files. Some cryptographic functions of the ICC V1.4.5 have been validated by Federal Information Processing Standard (FIPS) 140-2 for Security Level 1.

The TOE consists of:

- the GSKit SSL component, offering an API for SSLv3 and TLSv1 connections
- a key and a certificate management API and CLI
- the FIPS 140-2 level 1 validated cryptographic functions of the underlying IBM Crypto for C (ICC) Version 1.4.5 component.

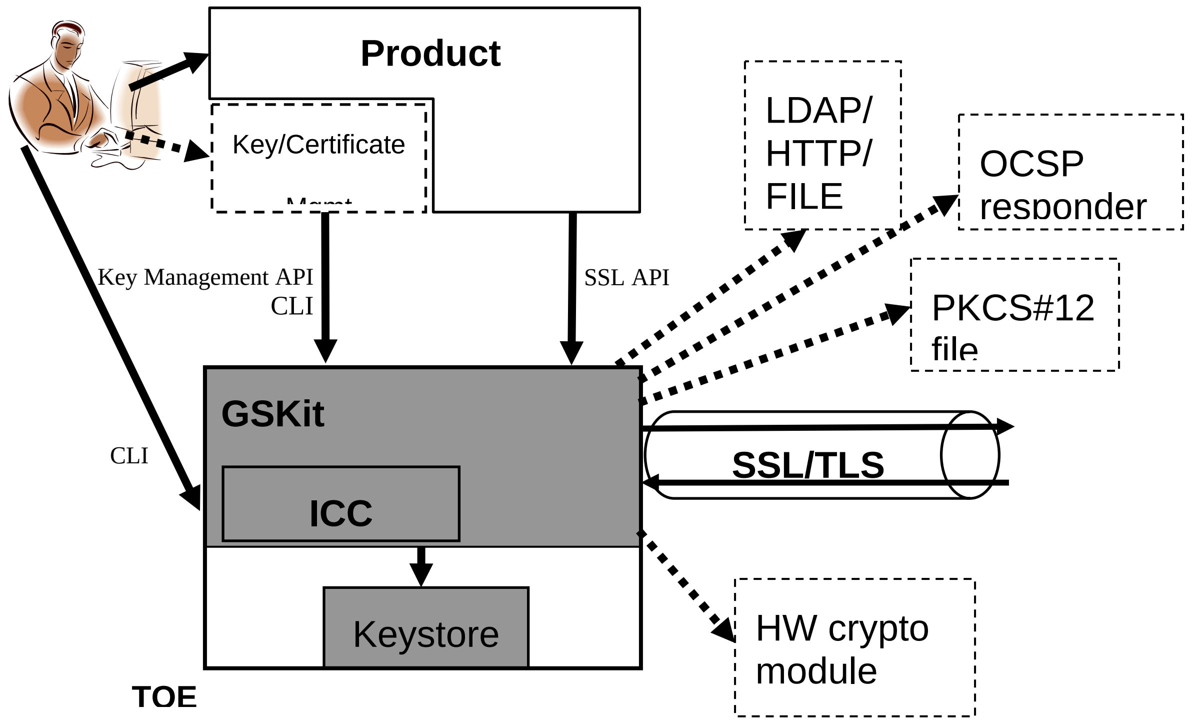
The non-FIPS validated/allowed cryptographic functions of ICC V1.4.5 – except for MD5in the SSLv3 and TLS cipher suites –, SSL versions prior to 3 (due to known weaknesses), the optional BSAFE cryptographic library, the optional iKeyman V7.0 component, optional hardware crypto modules, and the underlying operating system are not part of the TOE.

The operating system is not part of the TOE. It provides identification and authorization for TOE users, and a time source to be used by the TOE for certificate validation. The operating system is also responsible to protect the access to TOE services, the GSKit software, TSF data, and the keystore. As such, it provides significant support for the security of the TSF and TSF data: the TOE itself has only limited ability to protect against the corruption of TSF (i.e. the ICC module implements basic self-checks to satisfy FIPS 140-2 requirements). The underlying operating system is responsible for preventing bypass of the TSF or compromise of the TOE and its data by restricting access to authorized users only.

4.3. Availability of the Composition Requirements Definition (CRD)

IBM will make the CRD and the incorporated documents available upon formal request to integrators of GSKit who demonstrate their plan to integrate GSKit as a component into their TOE, provided that a proper business relationship and Confidentiality Disclosure Agreements are in place.

5. ARCHITECTURAL INFORMATION



GSKit is a library that can be used as a component within a larger product; it is intended to be integrated into such a product to provide this product with the capability to use an SSL- or TLS-protected communication channel to another product, providing authentication of the communication partners and protection against disclosure or undetected modification of the data transferred over this trusted channel. Furthermore, GSKit offers a command line interface which can be used for accessing key generation and management functions of GSKit, for example by a human user rather than by a program.

For (optional) client and (mandatory) server authentication, X.509 [X.509] certificates are used. These certificates can be imported from the environment. In addition to that, GSKit offers certificate generation and management functionality.

GSKit requires the product it is integrated into and the underlying operating system to provide the protection of the GSKit executables, security attributes and data, especially cryptographic keys and certificates. This includes a correct configuration of the TOE and that the processing resources of the TOE must be located within controlled access facilities.

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 TOE is provided in form of multiple binaries that have been compiled for the different operating systems supported by GSKit and the hardware architectures these operating systems run on (see

Section 8). The TOE does not interface hardware directly, but receives direct support for its security functions from the operating system. A direct dependency of the TSF on the underlying operating system is for the provision of a reliable time source for certificate validation.

6. DOCUMENTATION

The following documents form the TOE guidance:

- GSKCapiCmd User's Guide
- Global Security Kit Common Criteria Mode Operating Guidance
- IBM Global Security Kit Key Management for C Programmer's Guide
- Global Security Kit Install and Packaging Guide
- IBM Global Security Kit – Secure Socket Layer for C Programmer's Guide
- IBM Global Security Kit Trust Policy Design for GSKit
- Performing GSKit Local Installations
- Global Security Kit Delivery Procedure Additional Guidance

GSKit is only for IBM internal use and is not offered for sale as a standalone product, i.e., it is designed for the use in other IBM products. There is a formal process in place by which GSKit binaries, header files and documentation can be requested within IBM. After approval, permissions to access the TOE software and guidance over an IBM internal secured network connection are given and the TOE can be securely retrieved by the product team.

7. IT PRODUCT TESTING

7.1. Developer Testing

The developer's test environment for the TOE is comprised of the 20 systems listed in the Security Target (and in Section 8) as part of the evaluated configuration, plus some additional machines that were not relevant due to not being part of the evaluated configuration. In addition, the developer provided an LDAP server and an OCSP responder in the IT environment.

The developer employs a test strategy in which basic function testing is executed in an automated fashion after the build process. This testing serves as build verification testing and is not necessarily performed in the evaluated configuration. In addition, and as primary testing to achieve test coverage and depth for the evaluation, the developer maintains and uses automated and manual function verification tests (FVT). While partially derived from the build verification test suites, those tests allow execution in the evaluated configuration as described above. Tests are primarily implemented using the TOE's APIs. Direct test coverage is achieved for the complete TSFI; subsystem internal interfaces are primarily tested indirectly, with a smaller set of interfaces tested directly. For the cryptographic algorithms that have been FIPS 140-2 validated, the FIPS validation provides appropriate test coverage. The developer utilizes the Public Key Interoperability Test Suite (PKITS) developed by NIST for Certificate Path Validation to ensure that path validation is performed in accordance with the X.509 and the RFC3280 specifications.

The developer provides tests for all security functions identified in the TOE Summary Specification and Security Functional Requirements for the TOE as defined in the Security Target. The developer also tests the TSFI and the internal subsystem interfaces in sufficient detail.

A specific check was made to ensure that a test was in place to check for the known flaw in some implementations of RSA digital signatures using the padding scheme for RSASSA-PKCS1-v1_5 (as specified in Public Key Cryptography Standards (PKCS) #1 v2.1: RSA Cryptography Standard 2002) when the public key $e = 3$. It was verified that this condition is tested and that the GSKit code is not susceptible to the attack.

The evaluators ascertained that the testing was complete and comprehensive, covering explicit functionality as well as error conditions (e.g., invalid parameters, invalid credentials).

7.2. Evaluator Testing

As an integral component of testing, the evaluator installed and configured the following binary packages provided by the sponsor:

- GSKit binaries for Linux ia32 (32 bit Linux)
- GSKit binaries for Linux x86_64 (64 bit Linux)
- GSKit binaries for Windows (32 bit)

On the following systems for independent testing:

- The GSKit binaries for Linux x86_64 were installed on a SLES10 system running on an AMD 64-bit x86_64 CPU (execution of developer tests).
- The GSKit binaries for Linux ia32 were installed on a RHEL4 system running within VMWare on an Intel 32-bit Pentium-M CPU (execution of evaluator tests).
- The GSKit binaries for Windows were installed on a Microsoft Windows XP system running on an Intel 32-bit Core Duo CPU (execution of developer tests and evaluator tests).

The decision to test with the above configurations rather than all of the platforms and configurations specified in Section 8 was based on the following criteria:

- The Unix platforms use a common code base for the majority of the GSKit code. Some differences are surrounded by #ifdef and #endif directives. The majority of differences are related to low-level optimizations within the crypto related code. Since the crypto related code is subject to FIPS-2 certification, it is not significant to the testing associated with the evaluation.

- The code is written in ANSI C and C++ which minimizes the possibility of differences on different platforms.

These configurations matched supported platforms as specified in the Security Target, and were chosen to test both 32-bit and 64-bit CPUs running on Unix and Windows operating systems. All tests used the TOE in its evaluated configuration, as specified in the Security Target and the "Common Criteria Mode Operating Guide" as appropriate.

All security functions defined in the TOE summary specification were addressed. Emphasis was on the security functions related to key management, SSL/TLS secure channel establishment, and secure state preservation, since the majority of cryptographic algorithms implemented by the TOE have already been FIPS-validated. As a result, the subset size chosen covers about 60% of the TOE's security functional requirements

7.2.1. Independent Test

The sponsor's test suite was judged to be quite complete and comprehensive, and thus the evaluator needed to design relatively few additional tests. However, some additional test cases were developed and executed for key management, keystore integrity, SSL/TLS connection establishment including path validation and error handling, and correct handling of self-test errors received from ICC subsystem.

The following tests in the independent test suite were environment specific and thus the testing environment affected the test results:

Key data zeroization verification test. This test verifies that upon deallocation of memory buffer space, all key data gets cleared. The test requires a binary compiled with symbol information included so that a debugger can be used to stop and verify memory contents at specific points in the program execution. The test executed as expected on a linux-ia32 system. The test could not be executed on the Windows system due to the inability to generate debugging information in that environment, however the evaluator is confident that the functionality works as expected on the Windows platform since the developer has Build Verification Tests (BVT) that verify this.

Secure state on failure verification test. This test simulates a failure in the self-test routine, and verifies that the system properly handles the error and fails in a secure state. This test requires a GSKit binary compiled with symbol information included so that a debugger can be used to modify the expected value for the SHA-1 known answer test inside the ICC library data segment. Due to the modification, the self-test code detects a mismatch, reports the error and does not proceed with a connection attempt or other use of the crypto library. The test executed as expected on a linux-ia-32 system. The test could not be executed on the Windows system due to the inability to generate debugging information in that environment, however the evaluator is confident that the functionality works as expected on the Windows platform since the developer has Functional Verification Tests (FVT) that verify this.

7.2.2. Penetration Test

The selection of the test subset for penetration testing was drawn primarily from the TOE's vulnerability analysis presented by the sponsor and the Vulnerability Assessment ETR.. The evaluator chose the following area to be explored during penetration testing:

- Buffer overflows in the SSL protocol
- Keystore integrity error

- Signing of PKCS#10 certificate signing requests in MD2/MD5 formats.
- SSL connection with SSLv2 protocol
- Path validation of certificates

Tests were devised for each of the above areas and upon execution, no vulnerabilities were discovered.

8. EVALUATED CONFIGURATION[1]

The table below identifies the GSKit binaries that are part of the evaluated configuration, and for each of them the operating system versions they have been tested on and the additional operating system versions that the evaluation results can be extended to. For each GSKit binary, the operating system versions listed for a binary in the "Compatibility" column have been found to provide the necessary support functions in the same way as the operating system version that is actually used for testing during the evaluation.

TOE binary version and hardware	Compatibility
Sun Solaris 32-bit, running on Sparc and Ultrasparc	Solaris 7, 8, 9, 10
Sun Solaris 64-bit, running on Ultrasparc	Solaris 7, 8, 9, 10
HPUX 32-bit, running on PA-RISC1 and PA_RISC2 architectures	HPUX 11.0, 11i, 11iV2
HPUX 64-bit, running on PA-RISC2 architectures.	HPUX 11.0, 11i, 11iV2
IBM AIX 32-bit, running on PPC32 & PPC64 architectures	AIX 4.3.3, 5.1, 5.2, 5.3
IBM AIX 64-bit, running on PPC64 architectures	AIX 5.2, 5.3
Microsoft Windows 32-bit, running on IA32 architectures	Windows NT, XP, 2000, 2003, Vista, and future version
Microsoft Windows 64-bit, running on AMD x86_64 architectures	Windows 2003, Vista
RedHat Enterprise Linux 32-bit, running on IA32 architectures	UL 1.0 RHEL 2.1 AS, WS; RHEL 3.0 AS, ES, WS; RHEL 4.0 ES; RHEL 5.0 ES
RedHat Enterprise Linux 32-bit, running on	UL 1.0 RHEL 3.0 AS, ES, WS; RHEL 4.0

[1] For more complete information on the evaluated configurations, see Section 2.4.3 of the Security Target.

TOE binary version and hardware	Compatibility
PPC32 & PPC64 architectures	AS RH 5.0 ES
RedHat Enterprise Linux 32-bit, running on s390/zSeries architectures	UL 1.0 RHEL 3.0 AS; RHEL 4.0 AS
RedHat Enterprise Linux 64-bit, running on AMD x86_64 architectures	UL 1.0 RHEL 3.0 WS, AS; RHEL 4.0 WS, AS, ES; RHEL 5.0 ES
RedHat Enterprise Linux 64-bit, running on PPC64 architectures	RHEL 3.0 AS, ES, WS; RHEL 4.0 AS RH 5.0 ES
RedHat Enterprise Linux 64-bit, running on zSeries architectures	UL 1.0 RHEL 3.0 AS; RHEL 4.0 AS
SUSE Linux Enterprise Server 32-bit, running on IA32 architectures	UL 1.0 SLES 8, 9, 10
SUSE Linux Enterprise Server 32-bit, running on PPC32 & PPC64 architectures	UL 1.0 SLES 8, 9, 10
SUSE Linux Enterprise Server 32-bit, running on s390/zSeries architectures	UL 1.0 SLES 8, 9,10
SUSE Linux Enterprise Server 64-bit, running on AMD x86_64 architectures	UL 1.0 SLES 8, 9, 10
SUSE Linux Enterprise Server 64-bit, running on PPC64 architectures	SLES 8, 9, 10
SUSE Linux Enterprise Server 64-bit, running on zSeries architectures	SLES 8, 9, 10

9. RESULTS OF THE EVALUATION[2]

The evaluation team determined the product to be **CC Part 2 extended, CC Part 3 conformant**. In short, the product satisfies the security technical requirements specified in *IBM Global Security Kit Version 7.0.4.11 Security Target,* Version 1.7, 2007-07-26.

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

10. VALIDATOR COMMENTS / RECCOMMENDATIONS

10.1. Validation Comments

The Validation team observed the following:

When GSKit signs a certificate using “gsk7capicmd –cert –sign”, the resulting certificate file is stored in binary format (DER) even though the documentation claims the default is ASCII (base64). Explicitly specifiying the ‘–format ascii’ option on the command has no effect. GSKit will accept certificates in either format; however the resulting signed certificate is always in DER format.

GSKit permits nonce based requests and nonce-less requests. The validator recommends that the applications that require a response that is protected from replay attack should utilize nonce based request.

10.2. Validation Reccomendations

The Validation team observed that the evaluation and all its activities were performed in accordance with the CC, the CEM, and CCEVS practices. The evaluation was initiated prior to the establishment of the VOR process. As such, a Policy 10 and Policy 13 review was conducted with a result of pass. Since the VOR process was not in effect, an initial kickoff meeting was conducted rather than an Initial VOR. The Validation team agrees that the CCTL presented appropriate rationales to support the evaluation results and conclusions presented in the ETR. The Validation team therefore concludes that the evaluation and PASS result for this TOE are complete and correct for IBM Global Security Kit Version 7.0.4.11.

11. ANNEXES

11.1. Response to validator comments from previous evaluation

The following table details the vendors responses to the comments made by the validator during the previous evaluation.

Validator Comment	Proposed Mitigation	Long Term Recommendations	Vendor Resolution
The TOE does not properly process CRLs	The IT Environment must ensure that only full and complete CRL with no IDP are made available. The IT Environment must ensure that the latest CRL is made available.	The vendor shall fix the revocation processing in the next release.	Rectified: GSKit now preserves the STATUS or ‘UNDETERMINED’ for the calling application should an out of date CRL be presented and no fresh revocation information can be obtained from any source. CRL with IDP are supported correctly. This reported problem maybe due to using OpenSSL as the PKITS test harness.

Validator Comment	Proposed Mitigation	Long Term Recommendations	Vendor Resolution
The TOE does not process policies correctly	The TOE must not be used in environments where SSL/TLS connection is permitted on the basis of certificate policy. An example of this is cross certified environment.	The vendor shall fix the certificate policy processing in the next release.	Problem was due to using OpenSSL as the test harness for PKITS testing. No problem exists in GSKit itself. However a bug was found in the way GSKit processed Multiple Policy Set Intersections and this was rectified without impact to the NIST PKITS suite results. NIST PKITS suite results not using OpenSSL as test harness confirm correct operation.
The TOE does not process name constraints correctly	The TOE must not be used in environments where SSL/TLS connection requires name space constraints. Examples of this are cross certified environments and large global enterprises where name uniqueness is assured through name constraints.	The vendor shall fix the name constraints processing in the next release.	Problem was due to using OpenSSL as the test harness for PKITS testing. No problem exists in GSKit itself. NIST PKITS suite results not using OpenSSL as test harness confirm correct operation.
The TOE does binary name matching	Aside from denial of service, the only security problem this can cause is accept a certificate due to the same name encoded differently in excluded subtree. This was deemed acceptable to reduce the TOE complexity. The IT Environment must code all names the same ways (e.g., in terms of white spaces and lower and upper cases)	None	Problem was due to using OpenSSL as the test harness for PKITS testing. OpenSSL itself uses binary name matching whereas GSKit uses correct matching algorithm. No problem exists in GSKit itself. NIST PKITS suite results not using OpenSSL as test harness confirm correct operation.
The TOE does not enforce key usage extension when it not marked critical.	This is somewhat mitigated by ensuring that basic constraints must be present in the CA certificates	The vendor shall fix key usage extension processing in the next release.	Rectified. Problem is thought to be due to GSKit not being configured to use only the RFC 3280 validation engine (due to missing Guidance information) at the time of the test as the GSKit code was changed to accommodate this.
The TOE does not enforce key usage extension during CRL signature verification.	This is mitigated by the TOE requirement that the certificate and CRL be signed using the same key.	The vendor shall fix key usage extension processing in the next release.	Rectified. Problem no longer exists. Problem may have been due to using OpenSSL as test harness.

12. SECURITY TARGET

The ST, *IBM Global Security Kit Version 7.0.4.11 Security Target*, Version 1.7, 2007-07-26 is included here by reference.

13. LIST OF ACRYONYMS

API	Application Programming Interface
CC	Common Criteria
CCEVS	Common Criteria Evaluation and Validation Scheme
CCTL	Common Evaluation Testing Laboratory
CEM	Common Evaluation Methodology
CLI	Command Line Interface
CRL	Certificate Revocation List
EAL	Evaluation Assurance Level
ETR	Evaluation Technical Report
FIPS	Federal Information Processing Standard
NIAP	National Information Assurance Partnership
NIST	National Institute of Standards & Technology
NSA	National Security Agency
PP	Protection Profile
RSA	Rivest-Shamir-Adleman
SHA	Secure Hash Algorithm
SSL	Secure Sockets Layer
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Function

TSFI	TOE Security Function Interface

14. BIBLIOGRAPHY

[1] Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, dated August 2005, Version 2.3.

[2] Common Criteria for Information Technology Security Evaluation – Part 2: Security functional requirements, dated August 2005, Version 2.3.

[3] Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance requirements, dated August 2005, Version 2.3.

[4] Common Evaluation Methodology for Information Technology Security Evaluation – Evaluation Methodology, dated August 1998, version 2.3.

[5] Evaluation Technical Report, for the IBM Global Security Kit (GSKit) Version 7.0.4.11 with IBM Crypto for C (ICC) 1.4.5, Version 1.3, August 02, 2007.

[6] Global Security Kit Common Criteria Test Plan and Procedures (Proprietary); Version 0.7, February 27, 2007

[7] GSKit Test Depth Analysis (Proprietary); Version 1.00, May 27, 2007.

[8] IBM Global Security Kit v7.0.4.11 CC Test Plan (atsec Proprietary); Version 1.1, May 27, 2007

[9] IBM Global Security Kit Version 7.0.4.11 Composition Requirements Definition, Version 1.03, May 24, 2007

[10] IBM Global Security Kit Version 7.0.4.11 Security Target, Version 1.7, July 26, 2007

[11] Developer FVT Test Logs, May 23, 2007

[12] Developer test report summary, May 23, 2007