
Financial Services Security Laboratory



Master Security Criteria

Technical Contact Information

If further information regarding technical content is required, please contact:

Financial Services Security Lab
 c/o Global Integrity Corporation
 (888) 660-0134
 bitslab@globalintegrity.com
 BITS
 (202) 289-4322

Document Feedback

If you have any comments (technical or otherwise) regarding this document, please send an email to <mailto:bitslab@globalintegrity.com>. Include the document name, your name, e-mail address, telephone, and FAX number, and whether you would like to be contacted. *Please note ... BITS and Global Integrity will take all comments under advisement, but reserves the right to include or exclude comments received in the final document.*

Master Security Criteria – Document Version Control History

Note: **Bold** in Version column indicates a Public Release

Version / Date	Changes
1.0 (Aug 1999)	<ul style="list-style-type: none"> ◆ Initial Public Distribution
1.1 (Feb 2000)	<ul style="list-style-type: none"> ◆ <i>General</i>: Formatting improved ◆ <i>Page i</i>: “Document Feedback” section added; Document Version Control History table added ◆ <i>Pages 1</i>: 2nd paragraph added ◆ <i>Section 2.1.1 (Security Features, Identification)</i>: Eliminated bullets, now using outline numbers; Item 5 updated ◆ <i>Section 2.1.2 (Security Features, Authentication)</i>: Added new outline numbers to sub groups ◆ <i>Section 2.1.2.1 (Security Features, Authentication, General Mechanism Requirements)</i>: Item 1, 2 updated; Item 9 new/inserted ◆ <i>Section 2.1.2.2 (Security Features, Authentication, Knowledge-and-Possession-Based Mechanism Requirement)</i>: Item 4, 8 updated

Version / Date	Changes
	<ul style="list-style-type: none"> ◆ <u>Section 2.1.2.3</u> (<i>Security Features, Authentication, Personal Characteristics-Based Mechanism Requirements</i>): Bullet removed, using outline numbers ◆ <u>Section 2.1.3</u> (<i>Security Features, Authorization</i>): Items 13-14, 19 updated ◆ <u>Section 2.1.4</u> (<i>Security Features, Confidentiality</i>): Items 3, 5, 9-12 updated ◆ <u>Section 2.1.5</u> (<i>Security Features, Data Integrity</i>): Items 2, 4 updated ◆ <u>Section 2.1.6</u> (<i>Security Features, Audit</i>): Items 6, 9 updated; Item 13 new/inserted ◆ <u>Section 2.1.10</u> (<i>Security Features, Guidance</i>): Item 1 new/inserted ◆ <u>Section 2.1.11</u> (<i>Security Features, Non-Repudiation</i>): Item 1 updated; Item 4 new/inserted ◆ <u>Section 2.4</u> (<i>Scalability</i>): Bullet removed, using outline numbers

Table of Contents

1. INTRODUCTION.....	1
2. MASTER TEST CRITERIA	2
2.1 SECURITY FEATURES.....	2
2.1.1 <i>Identification</i>	2
2.1.2 <i>Authentication</i>	3
2.1.3 <i>Authorization</i>	6
2.1.4 <i>Confidentiality</i>	8
2.1.5 <i>Data Integrity</i>	9
2.1.6 <i>Audit</i>	10
2.1.7 <i>Data Disposal</i>	12
2.1.8 <i>System Integrity</i>	13
2.1.9 <i>Security Administration</i>	14
2.1.10 <i>Guidance</i>	15
2.1.11 <i>Non-Repudiation</i>	16
2.2 FUNCTIONALITY	16
2.3 USABILITY.....	16
2.4 SCALABILITY.....	17

1. Introduction

This document defines the master set of requirements that will support the technical analysis of a given product. A product will be tested within a standard configuration environment that may include the product and the supporting platform. The Master Criteria define the complete set of security features, functionality, usability and scalability requirements that apply to many different product categories. These criteria form the basis for the creation of product class profiles, which will be supported by test cases used to perform the actual technical analysis of a particular product. The criteria in the profiles will be applicable to the particular product class and the features and functions normally found in that product class.

The Master Criteria and product class profiles are focused on the implementation of specific product(s) in a prototypical business environment. Criterion outlined herein may be addressed through security features of underlying platforms, rather than the specific product [being tested] itself. Rather than requiring all security functionality to be provided by the standalone product (or system), the criteria and process allow a product to rely on an underlying platform (e.g., operating system, Web server) for security. To support this, the process allows the Technology Provider to define the “boundaries” of the test environment, which delineates the system to be tested. It is anticipated that this boundary will include the product itself, the underlying platform, and any other elements deemed appropriate by both the Technology Provider and the Financial Services Security Lab Manger. It is important to note, however, that the criteria will be applied equally to all components within that boundary.

2. Master Test Criteria

2.1 Security Features

Security features have been identified that outline the required system attributes to satisfy the security needs of a typical financial services organization and support its applications and infrastructure. The security features are viewed as generic, since they are not specific to any particular product or product class but provide baseline requirements that must be refined for a particular product. The requirements presented in this document are baseline in the sense that they are meant to be used as a starting point and a benchmark against which the actual attributes of a specific product can be measured through a testing process. The security features have been segmented into the following categories: Identification, Authentication, Authorization, Confidentiality, Data Integrity, Audit, Data Disposal, System Integrity, Security Administration, Guidance, and Non-Repudiation.

2.1.1 Identification

Identification is the process of recognizing a user's¹ unambiguous and auditable identity with the help of an *identifier* that is typically referred to as the user-ID. In general, the user-ID need not be confidential. It is the unambiguous name of a *user* through which the user can be held accountable². As such, all actions initiated by a user need to be associated with the corresponding user-ID. The security-related requirements in relation to user identification include the following:

1. The system shall unambiguously identify each user with the help of an identifier such as a user-ID.
2. Each interface³ that is accessed for system operations or to invoke services shall have the capability to recognize the user-ID.

¹ A user may be a person, a process, or a system that requests a session with the system to perform an operations- or services'-related task.

² A user may have multiple user-IDs as long as the multiple user-IDs unambiguously and uniquely identifies the user.

³ Typically, a system may have several different types of interfaces depending on the role of the user.

3. The system shall not allow an administrator to create, intentionally or inadvertently, a user-ID that already exists.
4. For each process running in the system that has been initiated by a user, the system shall associate the process with the user-ID of that user (e.g., if that activity is recorded in a history file, the record shall contain the corresponding user-ID). Autonomous processes (i.e., processes that are not initiated by a user, e.g., print spoolers, database management servers, etc.) shall be associated with an identifier code, such as “system ownership”.
5. The system shall have the capability to automatically disable⁴ an identifier if it remains inactive for a specifiable time period.
6. The system shall maintain the following list of security attributes for each user: user-ID, group memberships, access control privileges, authentication information and security-relevant roles.

2.1.2 Authentication

Authentication is the process of verifying the claimed identity of a user. Depending on the system and the application, different kinds of authenticators can include passwords, tokens, Smartcards, key-based authenticators, voice recognition, and/or a retina scan. Regardless of what type is used, it is critically important to minimize the compromise of an authenticator. Mechanism requirements have been divided into three categories: **General** applies to all types of authentication mechanisms; **Knowledge- and Possession-based** addresses mechanisms such as passwords; and **Personal Characteristics-based** provides guidelines for biometric mechanisms. Following are the security-related requirements for each type.

2.1.2.1 General Mechanism Requirements

1. The system shall store the information used for authentication in an encrypted form (or equivalent mechanism), using public and widely accepted algorithms or financial services industry standards.

⁴ The disabling process need not be automatic. For example, the system may generate an autonomous message for the administrator indicating that a user-ID has remained inactive for the specified period. It is expected that the administrator will disable the user-ID. However, an automatic disabling feature shall exist that the administrator may enable.

2. The authentication process shall protect the system from a replay attack.
3. Error feedback during the authentication procedure shall provide no information to the user other than “invalid” (i. e., it shall not reveal which part of the authentication procedure is incorrect).
4. The system shall have the ability to authenticate itself to the user and to other systems during session establishment.
5. The system shall have the ability to re-authenticate the user during an active session.
6. During system recovery, authentication information shall be recoverable without unauthorized disclosure or loss of data and system integrity.
7. The system shall not provide any mechanism to “null” the authentication information for a user-ID. No user-ID shall be allowed unauthenticated system access.
8. The system shall not allow any user to bypass the authentication process.
9. The system shall have the capability to use standard APIs to interface with common security and directory services.

2.1.2.2 Knowledge- and Possession-Based Mechanism Requirements

In the process of authentication, security information has to be exchanged to verify the user identity. This feature is focused on specific requirements for mechanisms that support security information known and possessed by the user and submitted for validation.

1. The system shall not divulge the authenticator (e.g., password, PIN number, token seed, Smartcard seed, etc.) of one user to any other user, including an administrator.
2. The authentication information shall not be displayed in clear text.⁵
3. The authentication information shall not be available in clear text to any other user, including an administrator.

⁵ For example, this implies that, during a login, a password shall not be echoed in clear text. Additionally, any occurrence of a clear text password, encryption key or other authentication information in the memory shall be overwritten immediately after use.

4. The system shall allow password and PIN number, as applicable, to be user-changeable at an administrator-defined interval.
5. If a password mechanism is used, then the system shall prompt the user to change the initial password and deny access if the user does not comply.
6. The system shall offer an authentication information-aging feature, so that users shall be required to periodically change authentication information.
7. Prior to the expiration of authentication information, the system shall provide notification to the user regarding the imminence of expiration.
8. At the time of an attempted password change, the system shall require re-authentication by the user, as well as re-confirmation of the new password.
9. There shall be a mechanism to prevent the reuse of the authentication information within an administrator-defined period. For example, when updating a password, a user shall be prevented from using a password that was used in the recent past.
10. If the system stays constantly logged onto another system on a long-term basis (e.g., continuous session), the system shall not impose mandatory authentication information aging for such interfaces.
11. The system shall prevent the use of trivial and predictable authenticators. For example, there shall be a configurable complexity requirement for passwords so that they cannot be easily guessed.
12. When password mechanisms are used, the system shall require that the password is configurable to administrator-specified characteristics for minimum password length, minimum alphabetic characters, and minimum numeric or special characters.
13. The system shall not enforce the condition of uniqueness on a password.⁶

⁶ The system shall not prevent a user from unknowingly choosing a password that is already being used by another user. Otherwise the existence of that password would be divulged.

2.1.2.3 Personal Characteristics-Based Mechanism Requirements

This type of authentication mechanism securely captures the physical characteristics (e.g., fingerprint) of the user and provides that data to the authentication process for validating the identity of the user.

1. For authentication based on the personal characteristics of the user, the system has to minimize the chance of a masquerade attack by an unauthorized user.

2.1.3 Authorization

The authorization feature is focused on the controls associated with establishment of a session with the system, invocation of operations- or services-oriented tasks, or the access of information while it is stored.

1. The system shall not allow access to the system resources without checking the assigned rights and privileges of the authenticated user. The system shall not allow access to any resource without invoking the authorization process.
2. During a login, the system shall allow the entire login sequence to be completed before providing any response.
3. If several consecutive incorrect login attempts are made, the system shall generate an alarm after an administrator-specifiable number of attempts. The maximum allowed is four attempts.
4. When the threshold for invalid consecutive attempts is reached, the system shall lock out the connection for an administrator-specified period of time.
5. At the time of login and accessing system resources, the system shall provide the capability to generate an authorized administrator configurable warning banner. The administrator shall have the capability to create a warning banner that conforms to corporate policy and complies with appropriate state and local laws.
6. Upon successful session establishment, the system shall display the date and time of the last successful login.
7. The system shall provide a “time-out” feature. This means that if during an active session there has not been any exchange of messages across the connection for an administrator-specified period of time, the system

shall lock out the connection and require a successful re-authentication to regain access.

8. If a session is interrupted by a disruption due to power failure, system crash, transmission problems, or is terminated by the user, the connection shall be dropped. The establishment of a new session will require the normal user identification, authentication and authorization.
9. The system shall have the capability to restrict session establishment based on time-of-day, day-of-week, calendar date of the login, and source of the connection.
10. The system shall have features to assign user privileges (i.e., access permissions) to user-IDs (not authentication information⁷).
11. The system shall provide an enforceable mechanism through which users can be segmented into roles (e.g., administrator), involving access to security features and other administrative functions.
12. The system shall have features to assign privileges to interfaces.
13. The system shall not allow unintended resource access to any user who has not established system access (i.e., a login with identification and authentication).
14. Unless a user has explicit permission to access a restricted resource, the system shall deny the access.
15. Unless a port has explicit permission to access a resource, the system shall deny the access to all users who log in to that interface.
16. The level of granularity of the resource control mechanism shall be such that any given user who has logged in to any given interface can be granted access or denied access to any specific resource (based on the user privilege and the interface privilege). Examples of resources can be a function (e.g., back-up operation), and data (e.g., files, fields). This covers both the operations and service related interfaces.
17. System privileges should be defined with appropriate scope limitations.

⁷ Assigning user privileges to authenticators may compromise their confidentiality.

18. The system shall have the capability to prevent access to potentially damaging commands (e.g., delete all files) from users who do not need to execute such commands on a regular basis and from interfaces that are not intended to be used for such commands.
19. The system shall have the capability to impose access control based on specific functions (e.g. Create, Read, Update, Delete, Execute, etc.).
20. The system shall provide the administrator specifiable capability to limit the number of concurrent logon sessions for a given user.
21. The system shall not offer any mechanism to bypass authorization restrictions.
22. The system shall not allow a less privileged user to spoof as a highly privileged user.

2.1.4 Confidentiality

The confidentiality protection feature is focused on protecting sensitive information from unauthorized disclosure while the information is being generated, stored, manipulated or forwarded.

1. The system shall have the capability to protect security-related sensitive information from unauthorized disclosure while it is stored and in transit.
2. The system shall have the capability to protect security-related, user-defined selected information from unauthorized disclosure while it is stored or in transit.
3. The system shall not allow any user without appropriate privilege to bypass the administrator-configured confidentiality mechanisms.
4. If keying material is generated and stored, the system shall provide secure key storage that is difficult to compromise through a logical or physical attack.
5. If keying material is generated, the system shall make use of a standard key generation algorithm that generates non-predictable values.
6. Only public and widely accepted or financial services industry standard encryption algorithms shall be supported by the system.

7. Products shall support multiple standard algorithms and key lengths to ensure appropriate levels of security. The administrator shall be able to configure the default algorithm and key length.
8. Certificate-based systems shall support X.509 v3 certificates.
9. The system shall have the capability to enforce the administrator-specified time period for the validity of keys, certificates, and related information for a particular use and/or user.
10. Once the administrator-specified time period for valid use of keys has expired, the system shall prevent further use of the key. If the system has to support key recovery, then the system has to provide for the authenticated and authorized access to the pertinent key.
11. The system shall have the capability to enforce the immediate revocation of a user and the associated key(s) when requested by the administrator.
12. The system shall support key recovery of all encryption keys by an authorized third party⁸.

2.1.5 Data Integrity

This feature is focused on preventing and detecting unauthorized modification of data that is associated with a user, the system itself, or the communications path.

1. The interface between the user and the system and among systems shall provide secure integrity checking capabilities.
2. The system shall have the capability to identify the originator of any information received from an untrusted network interface.
3. The system shall have the capability to propagate, when appropriate, the original user identifier to the destination.
4. The system shall provide mechanisms to detect and react to communication security violations in real-time, such as replay attacks that duplicate an authentic message.

⁸ Note that this criterion is intended to support recovery by the implementor of the system (e.g., a bank or service provider), as opposed to government or other third party.

5. The system shall provide mechanisms to preserve the integrity of protocol header information and user data.
6. The system shall support protocols that bind the integrity of sensitive information with the integrity of the associated protocol information.
7. The system shall have the capability to preserve the integrity of software and data remotely loaded into the system.
8. The system shall have the capability to protect the integrity of audit log records by generating integrity checks (e.g., checksums) when the log records are created.
9. The system shall not allow any user to bypass the administrator-configured data integrity controls.
10. The system shall have the capability to protect data integrity by performing data integrity checks such as:
 - ⇒ Proper rule checking on data updates;
 - ⇒ Verification of message authentication code (MAC), keyed Hash Message Authentication Code (HMAC) or digital signature;
 - ⇒ Adequate alert messages in response to potentially damaging commands before execution;
 - ⇒ Proper handling of duplicate and multiple inputs;
 - ⇒ Proper handling of securely generated encryption keying information;
 - ⇒ Proper handling of overflow conditions.

2.1.6 Audit

This feature has to provide adequate capabilities to investigate unauthorized activities after an event, so that the proper remedial action can be taken. This implies the recording of security-relevant events into an audit log that can be analyzed by the administrator.

1. The system shall maintain an audit log (e.g., a history file) that provides adequate information for establishing audit trails on security breaches (as part of post-mortem analysis) and user activity.

2. For each event recorded in the audit log, the system shall also record the identifier of the user (the user-ID) accountable for the event.
3. The audit log shall maintain the confidentiality of the authenticators (e.g., passwords) by excluding them from being recorded.
4. As a minimum, the audit log shall record events such as:
 - ⇒ All sessions established, including those established by alternate IDs if multiple IDs are supported;
 - ⇒ Invalid user authentication attempts;
 - ⇒ Unauthorized attempts to access resources (software, data, process, etc.);
 - ⇒ Administrator actions;
 - ⇒ Administrator disabling of audit logging;
 - ⇒ Events generated (e.g., commands issued) to make changes in users' security profiles and attributes;
 - ⇒ Events generated to make changes in the security profiles and attributes of system interfaces;
 - ⇒ Events generated to make changes in permission levels needed to access a resource;
 - ⇒ Events generated that makes changes in the system security configuration;
 - ⇒ Events generated that makes modifications of the system software;
 - ⇒ Events generated that make changes to access system resources deemed critical (as specified by the administrator).
5. For each recorded event mentioned above, the audit log, as a minimum, shall record:
 - ⇒ Date and time of the attempted event;
 - ⇒ User-ID of the initiator of the attempted event;
 - ⇒ Names of resources accessed;
 - ⇒ Host name;

- ⇒ Success or failure of the attempt (for the event);
 - ⇒ Event type.
6. The audit log shall be protected from unauthorized access, modification or deletion.
 7. If the audit log malfunctions, the system shall generate a real-time alarm.
 8. The system shall generate a real-time alarm for the impending failure (e.g., running out of storage space) of the audit feature.
 9. The system shall provide authorized access to a primary and secondary administrator role to enable the retrieval, printing, and copying (to some long-term storage device) of the contents of the audit log.
 10. The system shall provide an administrator with audit analysis tools to selectively retrieve records from the audit log to perform functions such as producing reports, establishing audit trails, etc.
 11. The audit log and its control mechanisms shall maintain integrity and completeness through system restarts.
 12. The system shall prevent unauthorized disabling of the audit function.
 13. The system shall have the capability to archive security related events.

2.1.7 Data Disposal

This feature is focused on protecting sensitive information from unauthorized recovery and subsequent disclosure from internal system memory and storage after authorized use.

1. The system shall have the capability to overwrite memory and storage that renders the information unrecoverable to prevent disclosure of sensitive information.
2. The system shall restrict the capability to overwrite memory and storage to an authorized user.
3. The system shall ensure that any previous information content of a resource is made unavailable upon the allocation of the resource for usage.

2.1.8 System Integrity

This feature is focused on the functional integrity of the system, including the controlled creation, installation and operation of the system software and data.

1. The system shall provide an administrator with the capability to monitor the state of availability of critical system resources (e.g., overflow indication, lost messages, and buffer queues).
2. The system shall prevent buffer overflow conditions that allow for unauthorized access.
3. For software and data created or modified in the system, the system shall provide an administrator with the capability to retrieve the user-ID, date and time associated with that creation or modification.
4. The system shall provide an administrator with the capability to perform integrity checks (e.g., synchronization points, checksums) on system data and software.
5. The system shall provide an administrator with the capability to perform secure recovery. The system shall provide an administrator with the capability to back-up and restore all security-relevant data, such as system configurations, user profiles, and access permissions. The system shall have the capability to check the integrity of security data read from a back-up file when performing a restore function.
6. The system shall retain the existing security parameters even after a restart or a recovery⁹.
7. The system shall provide adequate alert messages (e.g., "Do you really mean it?") in response to potentially damaging commands before execution.
8. An administrator shall have the capability to generate a status report detailing the values of the parameters and flags that affect secure operation of the system.

⁹ For example, user-IDs and passwords that have been assigned to the users shall not revert back to a vendor-delivered default such as system/system or admin/admin.

2.1.9 Security Administration

This feature is focused on the required system capabilities and parameters that must be available to the administrator to operate and manage the system in a secure manner.

1. The system shall have a mechanism such that the execution of security administration functions can be reserved only for an appropriate administrator role (i.e., all other users shall be denied this permission).
2. A system with one or more interfaces shall provide an administrator the capability to display all users currently logged on.
3. A system with one or more interfaces shall provide an administrator the capability to independently and selectively monitor (in real-time) the actions of any user currently logged on and lock out that user if necessary.
4. A system with one or more interfaces shall provide an administrator the capability to independently and selectively monitor (in real-time) the activities at a specified terminal, port, or network address, and lock out that input device if necessary.
5. The system shall provide an administrator the capability to create, retrieve, update or delete all security-related attributes of users, interfaces, and software and data elements.
6. The system shall provide an administrator the capability to *specify* all security parameters, such as individual user-IDs and passwords, password aging intervals, time-out intervals, various alarm conditions, access permissions, and text of the warning banner.¹⁰
7. The system shall provide an administrator with the capability to retrieve, copy, print, and upload an audit log, but shall *not* provide the capability to modify or delete the audit log.
8. The system shall provide the capability for the administrator role to be able to override vendor-provided security defaults.

¹⁰ The implication is that security parameters should not be hard-coded. Instead, they should be settable by the administrator by executing appropriate commands.

2.1.10 Guidance

This feature is focused on the assurance aspect of system security by supplementing the technical security capabilities with appropriate direction on securely configuring, operating and managing the system.

1. The system shall provide a “User Guide on Securing the System” or an appropriate and comparable document.
2. The system security administration guide shall contain:
 - ⇒ Cautions about functions and privileges that need to be controlled when running a secure facility;
 - ⇒ Administrator functions related to security, including adding or deleting a user, changing the security characteristics of a user, generation of keying material and the revoking of user related security parameters;
 - ⇒ Recommendations for setting the minimum access permissions on all files and commands;
 - ⇒ Guidelines on the consistent and effective use of the protection features of the system, how they interact, and how to securely generate a new system;
 - ⇒ Guidelines for retaining accountability tracking information for an administrator-specified period (e.g., 6 months);
 - ⇒ Procedures necessary to initially start the system in a secure manner;
 - ⇒ Procedures to resume secure system operation after any lapse in system operation;
 - ⇒ Documentation on the use of the audit tools:
 - Procedures for examining and maintaining audit logs;
 - Detailed audit record structure for each type of audit event;
 - Procedures for periodic back-up and deletion of audit logs;
 - Procedures for checking the amount of free storage space available for the log files.

2.1.11 Non-Repudiation

This feature is focused on the system's capabilities to prevent users from denying their actions in terms of receiving or sending data.

1. The system shall have the capability with authenticity and integrity to record information related to the reception of specific information from a user or another system.
2. The system shall have the capability to securely link received information with the originator of the information and other characteristics such as time and date.
3. The system shall have the capability to interface with a specified trusted third party to obtain keying material that will link the received information or request with a specific user.
4. The system shall have the capability to provide non-repudiation protection by performing the generation and/or verification of digital signatures.

2.2 Functionality

These criteria focus on the robust operation of the system's functions and ensure that a standard approach has been implemented for consistent security at a commercial level of strength.

1. The system shall use algorithms to support security features that are based on current standards.
2. The system shall provide security functions that are reliable and robust with a 99.99% uptime profile.
3. The system shall provide the security functions that are associated with the product class (e.g., firewall systems provide the ability to filter traffic based on a set of rules).
4. The system shall provide the security functions in a consistent manner.

2.3 Usability

These criteria focus on the ease of use of security features for different user roles and system interfaces.

1. The system shall provide security functions that operate with minimal impact on the user.

2. The system shall provide a clear and consistent interface (e.g., menu-driven) to facilitate the user and security administration roles.
3. The system shall provide a comprehensive user interface (i.e., the interface shall not leave out important operations).
4. The user feedback shall be adequate and timely (e.g., real-time) to prevent the user from entering incorrect data (e.g., typographical errors).
5. The system shall provide positive feedback to the administrator on the implemented security configuration of the system.
6. The system shall enable an administrator to configure individual users or groups of users with specific security characteristics.
7. The system shall securely recover all of the security settings and stored security parameters during the normal recovery operation.

2.4 Scalability

These criteria focus on the limitations of the system to continue to operate in a secure manner when the characteristics of the operating environment (e.g., number of users) change.

1. The system shall be able to continue to operate securely when various operating parameters increase or decrease. These operating parameters include number of users, active sessions, requests for security function support, generation and distribution of keying material, access control checks, and transaction.