
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 or +1.703.375.2100
 bitslab@globalintegrity.com
 BITS
 +1.202.289.4322

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"> ◆ <u>General</u>: Formatting improved ◆ <u>Page i</u>: “Document Feedback” section added; Document Version Control History table added ◆ <u>Pages 1</u>: 2nd paragraph added ◆ <u>Section 2.1.1 (Security Features, Identification)</u>: Eliminated bullets, now using outline numbers; Item 5 updated ◆ <u>Section 2.1.2 (Security Features, Authentication)</u>: Added new outline numbers to sub groups ◆ <u>Section 2.1.2.1 (Security Features, Authentication, General Mechanism Requirements)</u>: Item 1, 2 updated; Item 9 new/inserted ◆ <u>Section 2.1.2.2 (Security Features, Authentication, Knowledge-and-Possession-Based Mechanism Requirement)</u>: Item 4, 8 updated ◆ <u>Section 2.1.2.3 (Security Features, Authentication, Personal Characteristics-Based Mechanism Requirements)</u>: Bullet removed, using outline numbers

Version / Date	Changes
	<ul style="list-style-type: none"> ◆ <u>Section 2.1.3 (Security Features, Authorization)</u>: Items 13-14, 19 updated ◆ <u>Section 2.1.4 (Security Features, Confidentiality)</u>: Items 3, 5, 9-12 updated ◆ <u>Section 2.1.5 (Security Features, Data Integrity)</u>: Items 2, 4 updated ◆ <u>Section 2.1.6 (Security Features, Audit)</u>: Items 6, 9 updated; Item 13 new/inserted ◆ <u>Section 2.1.10 (Security Features, Guidance)</u>: Item 1 new/inserted ◆ <u>Section 2.1.11 (Security Features, Non-Repudiation)</u>: Item 1 updated; Item 4 new/inserted ◆ <u>Section 2.4 (Scalability)</u>: Bullet removed, using outline numbers
2.0 (Sep-Nov 2000)	<ul style="list-style-type: none"> ◆ <u>General</u>: Added rationale for each criterion, ◆ <u>General</u>: Renumbered Criteria: Section 2.1 now Section 2; 2.2 now Section 3; 2.3, Section 4, 2.5, Section 5. ◆ <u>Introduction</u>: Added discussion of functionality vs. features, also mapping of MSC to individual profiles. ◆ <u>Section 2 (Security features)</u>: Eliminated duplicate and redundant criteria; cleaned up wording on specific criteria. ◆ <u>Section 3 (Functionality)</u>: Rewrote criteria to reflect actual profile inclusion and testing process. ◆ <u>Section 4 (Usability)</u>: Section removed, criteria 3.4.1, 3.4.2, 3.4.5 deleted; criteria 3.4.3 and 3.4.6 moved to Security Administration; criteria 3.4.4 and 3.4.7 moved to System Integrity ◆ <u>Section 5 (Scalability)</u>: Rewrote section to reflect verification of vendor claims, etc.

Table of Contents

1	INTRODUCTION	1
1.1	OVERVIEW	1
1.2	CRITERIA FRAMEWORK	1
1.3	REQUIRED VS. DESIRED CRITERIA	2
1.4	SECURITY FEATURES VS. PRODUCT FUNCTIONALITY	2
1.5	PROFILE-SPECIFIC CRITERIA.....	3
1.6	SPECIFIC TEST CONDITIONS AND PRODUCT BOUNDARIES.....	3
2	SECURITY FEATURES.....	5
2.1	IDENTIFICATION.....	5
2.2	AUTHENTICATION	7
2.2.1	<i>General Mechanism Requirements.....</i>	<i>7</i>
2.2.2	<i>Knowledge- and Possession-Based Mechanism Requirements.....</i>	<i>9</i>
2.2.3	<i>Personal Characteristics-Based Mechanism Requirements.....</i>	<i>11</i>
2.3	AUTHORIZATION	12
2.4	CONFIDENTIALITY	15
2.5	DATA INTEGRITY.....	17
2.6	AUDIT	19
2.7	DATA DISPOSAL	23
2.8	SYSTEM INTEGRITY	24
2.9	SECURITY ADMINISTRATION	26
2.10	GUIDANCE.....	28
2.11	NON-REPUDIATION	29
3	PRODUCT FUNCTIONALITY	31
4	SCALABILITY	32
	APPENDIX A: INDUSTRY STANDARDS	33
	APPENDIX B: GLOSSARY OF TERMS.....	35

1 Introduction

1.1 Overview

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.

1.2 Criteria Framework

The framework is designed to be hierarchical, from defining the overall security attributes and features expected in all types of products, to the specific test cases developed to test compliance of a particular product. Areas to be examined include interfaces, security-relevant internal functions and operating environment. The framework comprises the following elements:

- ◆ **Master Security Criteria** – This document defines the overall security requirements and functions that are expected in all classes of products. It identifies the requirements for the major attributes (e.g., system integrity) and functions (e.g., traffic filtering).
- ◆ **Product Classes and Associated Product Security Profiles** - Products with similar functions and applications are grouped into a product class. Product Security Profiles are more specific security requirements applicable to that product class, but are derived from and consistent with the requirements in the Master Security Criteria.
- ◆ **Product Specific Security Test Plans and Test Scripts** – The actual testing strategy and supporting test cases are created for a product submitted for testing by the vendor. The testing is designed to measure compliance to the applicable Product Security Profile.

1.3 Required vs. Desired Criteria

Unless otherwise noted, products are required to meet all criteria listed within this document or a product profile in order to obtain the BITS-Tested Mark. However, in certain situations, the industry has identified a criterion that represents a desired feature/capability of a product, but has similarly recognized that it may not be feasible for a vendor to provide that feature at the present time. Similarly, some criteria have been identified as representing desirable features, but not ones required to inclusion in the financial industry's infrastructure. Each of these criteria has been identified within the relevant Product Profile as "desired," rather than "required." Products that do not meet the "desired" criteria will not be penalized for that fact. However, products that do meet "desired" criteria will be so recognized within the test report issued by the Lab Manager upon successful completion of a test.

1.4 Security Features vs. Product Functionality

It is important to note the distinction between security features of a product, as outlined in Section 2 of the Master Security Criteria, and product functionality, as outlined in Section 3. Often, it is difficult to identify the difference between features and functionality, especially when the product's primary functionality is related to security. Within the Financial Services Security Lab, "security features" is used to refer to the capabilities of a product itself to be secured (e.g., administrative interfaces, logging, authentication to the product, etc.). The "security features" section of the criteria is thus applicable to all products, regardless of the product class in which they fall, or the product profile against which they are being tested.

The "product functionality" section of the criteria refers to the primary functionality of the product and how it is affected by security. For products whose primary functionality is not security (e.g., applications, databases, operating systems, etc.), this section of criteria will test how that functionality is impacted by the security features of the product, as described in Section 2 of the Criteria. However, for those products whose primary functionality is security-related (e.g., authentication systems, network security products, authorization systems, etc.), the "functionality" criteria will address the main purpose of the product. In the cases of these product profiles, the "functionality" section of the criteria will often be as

detailed, if not more so, than the “security features” section. Furthermore, since the Product Profiles address a wide variety of products within a class, it is permissible for the Profile to contain functionality criteria specific to a “subclass” of products (e.g., the Authentication Systems profile might contain criteria specific to biometrics systems, smart cards, PKI, etc.).

1.5 Profile-specific Criteria

Criteria listed in a Product Profile will be distributed primarily into two sections: “security features” and “functionality,” in accordance with the description in Section 1.4 of this document. In each product profile, the section of criteria related to “security features” should be a direct subset of the Master Security Criteria, Section 2 (possibly with the addition of profile-specific rationale). It is permissible for a Product Profile to exclude a specific MSC “security features” criterion from the Profile, provided that sufficient rationale is provided to understand why that criterion is not included. However, there should be no “security features” criteria in a Product Profile that are not also contained in the MSC.

However, it is permissible for a Product Profile to contain “functionality” criteria that are not included in the MSC. Since functionality varies between product classes, it is infeasible for the Master Security Criteria to contain all possible criteria that could be applied to any product in any class. Thus, most “functionality” criteria in a Profile are likely to be specific to that class (or subclass, as described above) of products. For this reason, all “functionality” criteria in a Profile are required to include rationale as to why the criteria is included, as well as why it is required or optional.

1.6 Specific Test Conditions and Product Boundaries

The Master Security 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 and 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 BITS Security Lab Manager. It is important to note, however, that the criteria will be applied equally to all components within that boundary.

2 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 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:

<i>CRITERIA</i>	<i>RATIONALE</i>
1. <i>The system shall unambiguously and uniquely identify each user with the help of an identifier such as a user-ID.</i>	It is necessary to identify individual users to provide appropriate accountability for actions. Shared accounts inhibit the accountability and auditability of

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.

<i>CRITERIA</i>	<i>RATIONALE</i>
	actions on a system.
<p>2. <i>Each system interface³ that is accessed for system operations or to invoke services shall have the capability to recognize the user-ID.</i></p>	<p>Identification must be applied equally across all interfaces to a system. In the event that a “backdoor” exists through which access is granted with no identification, the security of the system is compromised.</p>
<p>3. <i>The system shall not allow an administrator to create, intentionally or inadvertently, a user-ID that already exists.</i></p>	<p>All user-IDs must be unique to ensure appropriate accountability and auditability of actions.</p>
<p>4. <i>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”.</i></p>	<p>For identification to properly provide accountability, specific actions must be tied to the initiator of that action. This includes long-term and short-term processes running on the system.</p>
<p>5. <i>The system shall have the capability to automatically disable an identifier if it remains inactive for a specifiable time period (e.g., three months).</i></p>	<p>Accounts that remain active, but dormant, are often the targets of attack. The longer that an account remains dormant, the greater the likelihood that it will be used for unauthorized purposes. 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</p>

³ The term “interface” refers to the point of entry into a system. It can be a network interface, user interface, or other system interface, as appropriate.

<i>CRITERIA</i>	<i>RATIONALE</i>
<p>6. <i>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.</i></p>	<p>user-ID. However, an automatic disabling feature shall exist that the administrator may enable.</p> <p>Identification must be tied to specific purposes to provide auditability of actions, accountability, authentication, and other security services.</p>

2.2 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.

2.2.1 General Mechanism Requirements

<i>CRITERIA</i>	<i>RATIONALE</i>
<p>1. <i>The system shall store the information used for authentication in an encrypted form, using public and widely accepted algorithms or financial services industry standards.</i></p>	<p>Authenticating information must be stored in such a way so that a third party without authorization to do so cannot easily obtain it. For example, static passwords should be passed through a one-hash function and only the hash should be stored.</p>
<p>2. <i>The authentication process shall protect the system from a replay attack by protecting the transmitted authentication information and examining sequences of</i></p>	<p>If transmissions can be eavesdropped, then it may be possible to replay the authenticator and convince the relying party that this is a new authentication attempt.</p>

CRITERIA	RATIONALE
<i>submitted authentication information.</i>	
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).	Feedback that is too descriptive can inadvertently give out information regarding which part of an authentication procedure is incorrect, thus allowing an attacker to narrow his or her search.
4. The system shall have the ability to authenticate itself to the user and to other systems during session establishment.	Too often, authentication focuses solely on the client authentication to the server. However, without proper server-to-client authentication, it may be possible for a third party to impersonate a server and obtain client’s authentication credentials.
5. The system shall have the ability to re-authenticate the user during an active session.	Periodic re-authentication improves a systems ability to withstand session “hijacking” attacks, in which a third party assumes control of a previously authenticated session.
6. During system recovery, authentication information shall be recoverable without unauthorized disclosure or loss of data and system integrity.	In the event of a system failure, it is possible for authentication information that was previously stored in volatile memory and not easily accessible, to be written to disk (e.g., as part of a core dump), allowing it to be retrieved by a third party.
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.	Each individual user must present the associated authentication information during the authentication process to ensure the authenticity and integrity of the user’s identity.
8. The system shall not allow any user to bypass the authentication process.	Each individual user must have associated authentication information to ensure the authenticity and integrity of the user’s identity.

2.2.2 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.

CRITERIA	RATIONALE
<p>1. <i>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.</i></p>	<p>In order for authentication information to properly identify the user with which it is associated, it is necessary to control the dissemination of that information. Administrators must have the ability to make changes to authentication information, but must not be able to easily impersonate the user.</p>
<p>2. <i>The authentication information shall not be displayed in clear text.</i></p>	<p>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.</p> <p>In the event that the information is displayed in the clear, it may be possible for a third party to impersonate the legitimate user.</p>
<p>3. <i>The authentication information shall not be available in clear text to any other user, including an administrator.</i></p>	<p>In the event that the information is transmitted in the clear, it may be possible for a third party to eavesdrop on the traffic, and thus impersonate the legitimate user.</p>
<p>4. <i>The system shall allow users to change their own password and/or PIN number at any time.</i></p>	<p>The system must allow for individual users to periodically make changes to their authentication information.</p>
<p>5. <i>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.</i></p>	<p>This criterion is aimed at preventing the user from relying solely on a default password that is known to third parties, such as an administrator.</p>
<p>6. <i>The system shall offer an</i></p>	<p>The periodic aging feature of authentication</p>

authentication information-aging feature, so that users shall be required to periodically change authentication information. This shall be administrator-configurable by user ID.

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.

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. The system shall prevent the use of trivial and predictable authenticators.

11. When password mechanisms are used, the system shall require that the password is configurable to administrator-specified characteristics

information serves to limit the time period during which a password/authenticator can be guessed, thus making it more difficult to crack passwords. In the event that a password/authenticator is compromised, aging of credentials serves to limit the potential use of, and damage caused by, that authenticator. However, in order to allow for ongoing communications between fully automated systems (e.g., batch processing), it is permissible for the system to not require aging under that specific circumstance.

In order to facilitate a user's changing of their authentication information; it is necessary to provide sufficient warning that a change is necessary. Failure to implement such warning may result in a user's inability to legitimately access their system.

In order to make a change to an authenticator, it is necessary for the user to be authenticated first, thereby preventing unauthorized changes to passwords.

As long as aging of authenticators is used, as described in criterion #6 above, it is necessary to place a minimum time interval on reuse of a password. Without this requirement, it is possible for users to alternate between two passwords periodically, making it easier for a third-party to obtain unauthorized access to the system, and defeating the purpose of the aging feature.

The use of trivial and predictable authenticators make it easier for a third-party to obtain an authenticator through brute-force attacks, such as dictionary attacks, and other cracking methods.

In conjunction with complexity requirements, it is necessary to enforce specific complexity requirements on passwords, as described herein.

for minimum password length, minimum alphabetic characters, and minimum numeric or special characters.

12. The system shall not enforce the condition of uniqueness on a password.

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. In the event that two users inadvertently choose the same password, the system should not disclose this fact to either party.

2.2.3 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.

CRITERIA	RATIONALE
1. For authentication based on the personal characteristics of the user, the system shall minimize the chance of a masquerade attack by an unauthorized user.	Biometric-based authentication conveys a certain amount of increased confidence in the system. Therefore, while it is always important to minimize the chance of masquerade attacks, it is particularly important in the case of biometric-based authentication systems, since they are perceived as having increased security.
2. The system shall support administrator-configurable parameters to control false reject and false accept errors.	Certain biometric systems are more susceptible to false positives than others. Since the tolerance level for false positives will undoubtedly vary between systems and organizations, it is necessary to have certain tunable parameters for acceptance/rejection of these cases.
3. The system shall take appropriate precautions to protect authentication information while stored. This includes the representation of the user's personal characteristics (e.g.,	Since it is difficult to forge a personal characteristic of an individual (e.g., a retinal pattern), the more common attack on a biometric system is against the representation of the biometric on the server. Similarly, since it is nearly impossible to reissue a

CRITERIA	RATIONALE
<i>fingerprint, iris pattern, etc.).</i>	biometric to a user, it is paramount that the representation of the biometric is adequately secured on the server.

2.3 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.

CRITERIA	RATIONALE
1. <i>The system shall not allow access to the system resources without checking the assigned rights and privileges of the authenticated user.</i>	Authorization is useless unless tied to something that maps identification to rights or privileges.
2. <i>The system shall not allow access to any resource without invoking the authorization process.</i>	Allowing one to bypass the authorization process could result in loss of data or system integrity. Authorization controls must be applied across all users, resources, and interfaces.
3. <i>During a login, the system shall allow the entire login sequence to be completed before providing any response.</i>	Providing response during the login process allows an attacker to determine which phase of the authentication process is incorrect (e.g., user-ID vs. password). This gives them an advantage towards future attacks.
4. <i>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.</i>	Multiple incorrect logins are often an indication of attempted intrusions. Allowing more than four incorrect attempts can dramatically decrease system security.
5. <i>When the threshold for invalid consecutive attempts is reached, the system shall lock out the account for an</i>	If multiple attempts occur, locking out the account can minimize the threat of unauthorized access, as well as allowing for time to perform forensic analysis

administrator-specified period of time.

6. At the time of login and accessing system resources, the system shall provide the capability to generate an administrator configurable warning banner. The administrator shall have the capability to create a warning banner that conforms to corporate policy and complies with appropriate national and local laws.

7. Upon successful session establishment, the system shall display the date and time of the last successful login.

8. 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 drop the connection and require a successful re-authentication to regain access.

9. 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.

10. 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.

11. The system shall have features to assign user privileges (i.e., access

of the incident.

Warning banners provide legal protection and policy awareness. Lack of the ability to create a warning banner can result in liability and lack of legal flexibility (e.g., with respect to prosecution of offenders). Similarly, since policies vary between organizations, it is necessary to allow for local customization of the warning banner.

Providing information about a user’s last successful login allows a user to determine the existence of an unauthorized login in the past, and is useful in detecting possible intrusions.

Leaving open active sessions increases the possibility of session hijacking as well as disclosure of data.

Disruptions and non-graceful shutdowns, if overlooked, leave open avenues for session hijacking and disclosure of data, as well as opportunities for unauthorized access to systems.

These features enable organizations to enforce more restrictive policies for non-standard access based on time and location.

Assigning user privilege to authenticators may compromise their confidentiality. Instead, assigning

- permissions) to user-IDs (not authentication information).*
- 12. 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.**
- 13. The system shall have features to assign privileges to interfaces.**
- 14. Unless a user has permission to access a 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 shall be defined with appropriate scope limitations.**
- privileges to user-Ids enables authorization checking without requiring disclosure of authentication.
- Providing for role-based access control allows individuals to have access based on a specific purpose, rather than just their identity. This minimizes the risk associated with providing superuser or other privileged access to individual users.
- Multiple interfaces to systems are often supported to separate operational access from administrative access. In order to support situations such as this, it is necessary to allow for specific privileges based on interface.
- By default, systems must take a “closed” approach to access control. This minimizes the possibility of “backdoor” unauthorized access to a system.
- By default, systems must take a “closed” approach to access control. This minimizes the possibility of “backdoor” unauthorized access to a system.
- Systems must ensure that user- and role-based access control, as well as interface-based access control are coordinated, thus allowing for maximum granularity of access control and minimizing the risk of inadvertent backdoor access.
- Many systems provide privileged access without any compensating controls or scope limitations, increasing the risk of damage from a rogue administrator or privileged user.

<p>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.</p>	<p>Limiting execution of damaging system commands can prevent damage (intentional or otherwise) to the system.</p>
<p>19. The system shall have the capability to impose access control on the basis of functions such as Create, Read, Update, and Delete (CRUD).</p>	<p>Access to system resources must be restricted in a manner that allows for granularity in accordance with organizational policy.</p>
<p>20. The system shall provide the administrator specifiable capability to limit the number of concurrent logon sessions for a given user.</p>	<p>In many situations, there may be a need to restrict concurrent logons for both security and consistency purposes. In some systems, concurrent logins can be used to mask unauthorized activity.</p>
<p>21. The system shall not allow a less privileged user to spoof as a highly privileged user.</p>	<p>Systems must enforce privileges in such a way that the integrity of access control mechanisms to highly privileged resources remains intact.</p>

2.4 Confidentiality

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

CRITERIA	RATIONALE
<p>1. The system shall have the capability to protect security-related sensitive information from unauthorized disclosure while it is stored and in transit.</p>	<p>Information related to security, if disclosed, can lead to unauthorized access to systems, disclosure of sensitive transaction and system information, and loss of integrity. It is equally important to protect data in storage as well as in transit from the system to other communicating entities.</p>
<p>2. The system shall have the capability to protect security-related, user-defined selected information from unauthorized</p>	<p>In addition to system-defined sensitive information, as covered above, it is important to protect user-defined sensitive information using similar</p>

CRITERIA	RATIONALE
<i>disclosure while it is stored or in transit.</i>	mechanisms.
3. The system shall not allow any user to bypass the administrator-configured confidentiality mechanisms.	The ability of users to bypass confidentiality mechanisms can dramatically decrease security of the system.
4. If cryptographic keys are generated and stored, the system shall provide secure key storage that is impractical to compromise through a logical or physical attack.	Keys provide the fundamental core to cryptographic protection, and their disclosure severely compromises confidentiality of data. Systems must provide key protection commensurate with the key's purpose.
5. If cryptographic keys are generated, the system shall implement a standard key generation algorithm that generates non-predictable values.	A cryptographic algorithm is only as strong as the strength of the key generation algorithm. These have proven to be the source of security breaches and vulnerabilities in the past.
6. Only public and widely accepted or financial services industry standard encryption algorithms, as described in Appendix A, shall be supported by the system.	Algorithms as identified in Appendix A have been exposed to public scrutiny and have been shown to meet the needs of the financial industry. Use of other algorithms may not provide the same level of security, and will likely not engender the same confidence in 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.	Required strength of mechanism varies based on the application and the environment. Since environments can vary greatly between organizations, the system must allow for local customization of the cryptographic mechanism.
8. Certificate-based systems shall support X.509 v3 certificates.	X.509 ⁴ is the accepted standard for digital certificates, and must be supported to promote interoperability between systems.
9. The system shall have the capability to enforce the administrator-specified time period for the validity of keys for a particular	The duration of key validity is directly proportional to the risk of compromise. Key validity periods also provide the ability to enforce local policies

⁴ Reference: ITU-T Rec. X.509(97) The Directory: Authentication Framework

<i>CRITERIA</i>	<i>RATIONALE</i>
<i>use and/or user.</i>	and procedures.
<i>10. Once the administrator-specified time period for valid use of keys has expired, the system shall prevent further use of the key.</i>	Key validity periods must be enforced by the relying party and at the point of issuance of the certificate. Due to caching and the vagaries of individual copies, it is not sufficient to enforce key validity solely at the sending party's end.
<i>11. If the system has to support key recovery, then the system has to provide for the authenticated and authorized access to the pertinent key.</i>	Recovery of keys should only be allowed by authorized parties.
<i>12. The system shall have the capability to enforce the immediate revocation of a user and the associated keying material when requested by the administrator.</i>	Revocation is necessary to enforce policy. Since all keys can be cached locally, the only way to prevent use of an unauthorized key or certificate is to globally denote revocation of that key or certificate.
<i>13. The system shall support authorized recovery of all encryption keys.</i>	It must be possible to recover a key in the event that the key's owner or primary caretaker is unable to provide the key for legitimate use (e.g., the key has been lost; the individual has left the company; etc.). This is in line with most financial institution's policies regarding corporate access to proprietary data.
<i>14. Signing keys shall not be used for purposes of data encryption.</i>	In conjunction with the requirement for key recovery in #13, above, it is necessary to separate encryption keys, which must be recoverable, from signing keys, which identify the owner and must not be recoverable.

2.5 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.

CRITERIA	RATIONALE
1. <i>The interface between the user and the system and among systems shall provide secure integrity checking capabilities.</i>	It is necessary to ensure the validity of transmission between systems (i.e., to ensure that the data received is the same as that which was sent.) In the event that the communication is over a network or via a location in memory or disk, this can be provided through various tools. If the data is on the same system, then a protected path is required.
2. <i>The system shall have the capability to identify the originator of any information received from a network interface or entered via the user interface.</i>	To support forensic analysis, it is necessary to be able to identify, to a reasonable certainty, the identity of the sender of a piece of data. This identification may take the form of the IP address of the sending machine, but is more appropriately tied to a user ID.
3. <i>The system shall have the capability to propagate, when appropriate the original user identifier to the destination.</i>	The source identification should be available to further, back-end systems, for audit purposes.
4. <i>The system shall provide mechanisms to detect communication security violations in real-time, such as replay attacks that duplicate an authentic message.</i>	Serious attacks on data integrity, such as replay, represent a significant threat to system security and need to be dealt with as soon as possible.
5. <i>The system shall provide mechanisms to preserve the integrity of protocol header information and user data.</i>	Data integrity is not limited merely to message content, but should also be applied to general packet header information and routing information.
6. <i>The system shall support protocols that bind the integrity of sensitive information with the integrity of the associated protocol information.</i>	Integrity of network data (e.g., protocol headers) should be tied to integrity of packet data, to further ensure the integrity of the transmission.
7. <i>The system shall have the capability to protect the integrity of audit log records by generating integrity checks (e.g., checksums or secure hashes) when the log records are created, and verifying the</i>	A common technique, as part of an attack, is to alter the log and audit records on a system to hide unauthorized activity. Integrity checks on these records can help prevent such activity.

CRITERIA	RATIONALE
<p><i>integrity check data when the record is accessed.</i></p> <p>8. <i>The system shall not allow any user to bypass the administrator-configured data integrity controls.</i></p> <p>9. <i>The system shall have the capability to protect data integrity by performing data integrity checks such as:</i></p> <p><i>Proper rule checking on data updates;</i></p> <p><i>Verification of message authentication code (MAC), keyed Hash Message Authentication Code (HMAC) or digital signature;</i></p> <p><i>Adequate alert messages in response to potentially damaging commands before execution;</i></p> <p><i>Proper handling of duplicate and multiple inputs;</i></p> <p><i>Proper handling of securely generated encryption keying information;</i></p> <p><i>Proper handling of overflow conditions.</i></p>	<p>Integrity controls must be uniformly applied to all users, resources, data, and interfaces.</p> <p>Data integrity is a large issue, and threats take many forms. The system should take steps like those listed herein to ensure that the integrity of data is maintained at all relevant points within a system.</p>

2.6 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.

CRITERIA	RATIONALE
<p>1 <i>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.</i></p>	<p>The event log is one of the strongest tools in forensic analysis, and can also help determine the root cause of an incident. Additionally, event logs help identify patterns of attack and abnormal behavior.</p>
<p>2 <i>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.</i></p>	<p>It is necessary to record the user ID with the event, in order to ensure auditability of activities.</p>
<p>3 <i>The audit log shall maintain the confidentiality of the authenticators (e.g., passwords) by excluding them from being recorded.</i></p>	<p>Logs are often visible to persons that are not similarly authorized to view authentication information. In order to enforce both the confidentiality of authentication data, as well as the separation of duties requirement by most financial institutions, it is necessary to exclude authentication information from audit logs.</p>
<p>4 <i>As a minimum, the audit log shall record events such as:</i></p> <p>4.1 <i>All sessions established</i></p> <p>4.2 <i>Failed user authentication attempts;</i></p> <p>4.3 <i>Unauthorized attempts to access resources (software, data, process, etc.);</i></p> <p>4.4 <i>Administrator actions;</i></p> <p>4.5 <i>Administrator disabling of audit logging;</i></p>	<p>All of these events have been deemed to be security-related, since they involve system access, security administration, or event logging. Each of these events can have significant impact on system security and, if unauthorized, are often signs of an attempted attack or intrusion.</p>

CRITERIA	RATIONALE
<p>4.6 <i>Events generated (e.g., commands issued) to make changes in users' security profiles and attributes;</i></p>	
<p>4.7 <i>Events generated to make changes in the security profiles and attributes of system interfaces;</i></p>	
<p>4.8 <i>Events generated to make changes in permission levels needed to access a resource;</i></p>	
<p>4.9 <i>Events generated that make changes to the system security configuration;</i></p>	
<p>4.10 <i>Events generated that make modifications to the system software;</i></p>	
<p>4.11 <i>Events generated that make changes to system resources deemed critical (as specified by the administrator).</i></p>	
<p>5 <i>For each recorded event mentioned above, the audit log, as a minimum, shall record:</i></p>	<p>In order to properly analyze the impact of an event, it is necessary to have this basic information for use during forensic analysis.</p>
<p>5.1 <i>Date and time of the attempted event;</i></p>	
<p>5.2 <i>User-ID of the initiator of the attempted event;</i></p>	
<p>5.3 <i>Names of resources accessed;</i></p>	
<p>5.4 <i>Host name;</i></p>	
<p>5.5 <i>Success or failure of the attempt (for the event);</i></p>	

CRITERIA	RATIONALE
5.6 <i>Event type.</i>	
6 <i>The audit log shall be protected from unauthorized access, modification or deletion. This protection shall be provided by assigning appropriate resource access permission to users and interfaces.</i>	Since the audit log is important for reconstructing historical activities, it is necessary to protect it from modification. Access to log files should be limited to those with a legitimate business need.
7 <i>If the audit log malfunctions, or is shutdown for any reason, the system shall generate a real-time alarm and have the capability to send an e-mail notification.</i>	If the audit log malfunctions, there is no way of tracking activities or transactions. This makes it difficult to conduct business under normal operations. Rather than shutting down the system, however, most organizations choose to operate under some form of emergency circumstances. In order to inform the organization of the need to invoke these new operational procedures and to begin to address the problem at hand, there is a need to sound a real-time alarm and send an e-mail notification (since real-time alarms to the console are typically ignored by the operators.)
8 <i>The system shall generate a real-time alarm and send an e-mail notification to the administrator for the impending failure (e.g., running out of storage space) of the audit feature.</i>	Audit logs most often fail due to a lack of storage space. Prior to shutting down or failing for this reason, the system needs to notify administrators so that precautions can be taken to ensure the availability of the audit service.
9 <i>The system shall provide the administrator the ability to retrieve, print, and copy (to some long-term storage device) the contents of the audit log.</i>	It is desirable to move audit logs to remote systems or storage in order to avoid system failure due to space restrictions, as well as to avoid unauthorized modification. Secondary administrator / operator / auditor roles are needed to allow for log off-loading and management without granting broader superuser privileges.
10 <i>The system shall provide an administrator with audit analysis tools to selectively retrieve records from the audit log to perform functions such as</i>	Audit logs quickly grow very large, making it difficult to get information out of them. This can result in a lack of utility of the log, as well as administrator frustration. Log reduction, search,

CRITERIA	RATIONALE
<i>producing reports, establishing audit trails, etc.</i>	and retrieval tools provide an easy way to get at relevant data.
11 <i>The audit log and its control mechanisms shall maintain integrity and completeness through system restarts.</i>	System restarts must not clear the log. Since a shutdown or restart can be associated with a security-related event (or even be one on its own), it is necessary for logs to note them as an event, and to resume normal operations upon restart.
12 <i>The system shall prevent unauthorized disabling of the audit function.</i>	It is recognized that under certain circumstances it may be necessary to disable audit logs to maintain critical system performance. However, since this is a drastic measure, the ability to do so must be limited to authorized personnel.

2.7 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.

CRITERIA	RATIONALE
1. <i>The system shall have the capability to overwrite memory and storage that renders the information unrecoverable to prevent disclosure of sensitive information.</i>	When memory or storage space is de-allocated, vestiges of previous data stored there can persist. Thus, upon reallocation of that space, that data may be made available, often to a different, unauthorized user. In order to prevent this, it is necessary to overwrite these areas of memory / disk prior to their reuse.
2. <i>The system shall restrict the capability to overwrite memory and storage to an authorized user.</i>	Since the ability to overwrite data can render the data unavailable, even to legitimate users during normal operations, the ability to overwrite must be restricted.
3. <i>The system shall ensure that any previous information content of a resource is made unavailable upon the allocation of the</i>	To ensure that data doesn't persist as described above, it is needed to ensure that the storage or memory space is clean upon reallocation of data

<i>CRITERIA</i>	<i>RATIONALE</i>
<i>resource for usage.</i>	(even if it wasn't overwritten upon de-allocation).

2.8 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.

<i>CRITERIA</i>	<i>RATIONALE</i>
<i>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).</i>	In order to properly ensure that a system is functioning properly, it is necessary for administrator to be able to monitor system metrics that affect normal operations.
<i>2. The system shall prevent buffer overflow conditions that allow for unauthorized access.</i>	Buffer overflow conditions are a common source of unauthorized access to system, allowing for an unauthorized user to cause a system to execute a (potentially damaging) command.
<i>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.</i>	In order to ensure that no unauthorized modifications are made to the system, and to support forensic analysis of suspected events, it must be possible for an administrator to track the source of individual events such as this.
<i>4. The system shall have the capability to preserve the integrity of software and data remotely loaded into the system.</i>	If integrity information surrounding third-party data is provided, the system should support/maintain the use of that information. If not, the system should check the third-party data to ensure integrity. Remote downloads can be the source of risk introduction to a system.
<i>5. The system shall provide an administrator with the capability to perform integrity checks (e.g., synchronization points, checksums) on system data and software.</i>	To proactively prevent against unauthorized system modification, the administrator must be able to conduct periodic checks on the system data and software's integrity.

CRITERIA	RATIONALE
<p>6. <i>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.</i></p>	<p>In the event of a system crash or shutdown, system performance becomes unpredictable. It is always necessary and imperative that the administrator be able to reconstruct the system according to security policy and good practices. Therefore, it is necessary to have a trusted security “baseline” configuration from which to restore.</p>
<p>7. <i>The system shall have the capability to check the integrity of security data read from a back-up file when performing a restore function.</i></p>	<p>In the event of a system crash or shutdown, system performance becomes unpredictable. It is always necessary and imperative that the administrator be able to reconstruct the system according to security policy and good practices. Therefore, it is necessary to have a trusted security “baseline” configuration from which to restore.</p>
<p>8. <i>The system shall retain the existing security parameters even after a restart or a recovery.</i></p>	<p>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.</p>
<p>9. <i>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.</i></p>	<p>In order to properly maintain the system and its performance from both an operational and security perspective, it is necessary to determine the current security “snapshot” of the system.</p>
<p>10. <i>The user feedback shall be adequate and timely (e.g., real-time) to prevent the user from entering incorrect data (e.g., typographical errors).</i></p>	<p>The system must perform in a way that mitigates and avoids security breaches. Security breaches due to a user erroneously responding to misleading or insufficient system information or information that may lead the user to respond in an erroneous way must be prevented by the system.</p>
<p>11. <i>The system shall securely recover all of the security settings and stored security parameters during the normal recovery operation.</i></p>	<p>When the system supports recovery of security settings across system invocations, then the benefits are efficiency and consistency. Consistency of security settings is particularly</p>

<i>CRITERIA</i>	<i>RATIONALE</i>
	important because change is always a point when errors may occur. When the system does the security setting recovery then the administrator is not placed in a configuration management situation that may lead to the errors that cause security breaches.

2.9 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.

<i>CRITERIA</i>	<i>RATIONALE</i>
<i>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).</i>	As with other security control mechanisms, the authorization to perform certain functions is often limited to a small set of people. By default, the system should take a closed security approach to all other users.
<i>2. A system with one or more interfaces shall provide an administrator the capability to display all users currently logged on.</i>	This allows the administrator to determine who currently has access to the system, and assists the administrator in identifying unauthorized users currently on the system.
<i>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.</i>	This provides the administrator the capability to monitor actions on the system, in support of the ability to monitor unauthorized activity. This capability is often used in “honeypots” and other forensic activity.
<i>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.</i>	Just as the administrator requires the ability to monitor activities of a specific user, the same is true of activities of a specific connection.

CRITERIA	RATIONALE
<p>5. <i>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.</i></p>	<p>In order to provide the greatest amount of granularity of security controls, an administrator must have the capability to change specific attributes of users and resources that are related to security.</p>
<p>6. <i>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.</i></p>	<p>The implication is that security parameters should not be hard-coded. Instead, they should be configurable by the administrator by executing appropriate commands.</p>
<p>7. <i>The system shall provide the capability for the administrator role to be able to override vendor-provided security defaults.</i></p>	<p>Vendor-provided defaults may not always meet the security requirements and policies of specific organizations. As some financial institutions may choose to impose organization-specific policies that may be more stringent than those enforced by the vendor, it is necessary to allow administrators to override defaults.</p>
<p>8. <i>The system shall provide a comprehensive user interface (i.e., the interface shall not leave out important operations).</i></p>	<p>For example, user operations could be made available through some script, batch, file or the command line instead of the primary user interface. This is contrary to the requirement of this criterion and is undesirable. The user interface should provide an easy-to-use, fully functional and comprehensive entry point to the operation of the system. Operations should be available via both the command line (scripts or otherwise) as well as the primary user interface.</p>
<p>9. <i>The system shall enable an administrator to configure individual users or groups of users with specific security characteristics.</i></p>	<p>In order to efficiently establish specific security characteristics, the system must have an interface and tools that enable the flexible administration of user attributes for all user classes and groups.</p>

2.10 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.

<i>CRITERIA</i>	<i>RATIONALE</i>
<i>The system security administration guide shall contain:</i>	In most cases, there are certain features of a system that can dramatically improve or harm security of the system, yet are not readily apparent to users or administrators. It is the responsibility of the vendor to notify system users and/or administrators, as appropriate, of these features.
1. <i>Cautions about functions and privileges that need to be controlled when running a secure facility;</i>	Certain systems may be more susceptible to vulnerabilities from specific users and/or privileges than others.
2. <i>Administrator functions related to security, including adding or deleting a user, changing the security characteristics of a user, generation of cryptographic keys and the revoking of user related security parameters;</i>	All security-related administration functions must be documented, so that these functions can be appropriately used to enhance the security posture of the system.
3. <i>Recommendations for setting the minimum access permissions on all files and commands;</i>	As access permissions on system files, commands, and other functions can vary from system to system, and that there is no single default minimum for all possible systems, it is incumbent upon the vendor to provide minimum security configuration settings in order to comply with the remainder of the criteria in this document.
4. <i>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;</i>	Often, various security (and other) system functions can adversely interact with each other. Alternatively, certain features can strengthen security when used in certain combination with other system features.
5. <i>Guidelines for retaining accountability tracking information for an administrator-specified period (e.g., 6</i>	In order to comply with audit log (and other) retention information, it may be necessary to configure the system in order to make information

<i>CRITERIA</i>	<i>RATIONALE</i>
<i>months);</i>	available.
<i>6. Procedures necessary to initially start the system in a secure manner;</i>	In the event that any non-standard activities (e.g., actions not normally required during system install or startup) are required to securely initiate system operations, they must be documented.
<i>7. Procedures to resume secure system operation after any lapse in system operation;</i>	Any specific actions necessary to support secure recovery and render the system compliant with system integrity and audit requirements must be documented.
<i>8. Documentation on the use of the audit tools:</i>	All actions necessary to support proper use of the audit log must be documented, to ensure that audit information is accurate and useful.
➤ <i>Procedures for examining and maintaining audit logs;</i>	
➤ <i>Detailed audit record structure for each type of audit event;</i>	
➤ <i>Procedures for periodic back-up and deletion of audit logs;</i>	
➤ <i>Procedures for checking the amount of free storage space available for the log files.</i>	

2.11 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.

<i>CRITERIA</i>	<i>RATIONALE</i>
<i>1. The system shall have the capability to securely record information related to the reception of specific information from a user or another system.</i>	In order to properly enforce non-repudiation, it is necessary to properly document receipt of all data.

<i>CRITERIA</i>	<i>RATIONALE</i>
<p>2. <i>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.</i></p>	In order to enforce non-repudiation, it is necessary to tie specific data to a user or system, as well as to the time at which it was sent. This supports accountability of actions, and is a core concept of non-repudiation.
<p>3. <i>The system shall have the capability to interface with a specified trusted third party to obtain cryptographic keys that will link the received information or request with a specific user.</i></p>	Digital signatures are often used to provide non-repudiation. In order to properly verify a signature, it may be necessary to obtain verification from a third party, such as a certification authority, validation authority, or certificate directory.

3 Product Functionality

This section of the criteria refers to the primary functionality of the product and how it is affected by security. For products whose primary functionality is not security (e.g., applications, databases, operating systems, etc.), this section will test how that functionality is impacted by the security features of the product, as described in Section 2 of the Criteria. However, for those products whose primary functionality is security-related (e.g., authentication systems, network security products, authorization systems, etc.), the “functionality” criteria will address the main purpose of the product. In the cases of these product profiles, the “functionality” section of the criteria will often be as detailed, if not more so, than the “security features” section. Furthermore, since the Product Profiles address a wide variety of products within a class, it is permissible for the Profile to contain functionality criteria specific to a “subclass” of products (e.g., the Authentication Systems profile might contain criteria specific to biometrics systems, smart cards, PKI, etc.).

<i>CRITERIA</i>	<i>RATIONALE</i>
<p><i>1. The main product functionality, as described in the Testing Services Agreement, shall not be substantially affected by the introduction of capabilities that enable the product to meet the security features criteria outlined in Section 2.</i></p>	<p>If the product meets the security features criteria, but ceases to function as originally intended, or the product’s ability to function is substantially lessened by the presence of security capabilities, this may render the product unusable in a production environment.</p>
<p><i>4. The Product shall comply with all required functionality criteria as outlined within the Product Profile</i></p>	<p>Since the functionality of products varies between classes, all product functionality criteria all included in the individual Profiles.</p>

4 Scalability

<i>CRITERIA</i>	<i>RATIONALE</i>
<p><i>1. The system shall be able to continue to operate securely when various operating parameters increase or decrease. These operating parameters shall be specified by the Technology Provider in the Product Test Schedule, and will be itemized in the Test Plan.</i></p>	<p>Since performance of products varies greatly from one environment to the next, and since scalability can be measured with respect to a number of parameters, the only scalability tests conducted under this framework will be a verification of vendor claims regarding the scalability of the product, and that security features (and, if applicable, functionality), remain consistent at those levels.</p>

Appendix A: Industry Standards

For the purposes of these criteria, the terms “public and widely-used or financial industry standards” shall refer to the following items, as specified within this document.

Symmetric Encryption algorithms	<ul style="list-style-type: none"> • 3DES (ANS X9.52, X9.66) • IDEA • RC4 • RC5 • RIPEM
Asymmetric Algorithms (for symmetric key agreement or key transport)	<ul style="list-style-type: none"> • RSA (ANS X9.44) • D-H (minimum 1024-bit modulus – ANSI X9.42) • ECDH (ANS X9.63)
Digital hashing algorithms	<ul style="list-style-type: none"> • SHA-1 (ANS X9.30-2) • MD5
Digital signature algorithms	<ul style="list-style-type: none"> • DSA (ANS X9.30-1) • rDSA (ANS X9.31) (includes RSA) • EC-DSA (ANS X9.62)
Key management standards and protocols	<ul style="list-style-type: none"> • ANS X9.70, ANS X9.73, ANS X9.69, ANS X9.24, ANS X9.77 • CMP • PKCS #7, #10 • IETF PKIX standards
Random number generators	<ul style="list-style-type: none"> • ANS X9.82
Prime number generators	<ul style="list-style-type: none"> • ANSI X9.80
Cryptographic Device Security	<ul style="list-style-type: none"> • ANS X9.66 • FIPS 140-2
Peer Entity Authentication	<ul style="list-style-type: none"> • ANS X9.72 • FIPS 196
PIN Security	<ul style="list-style-type: none"> • ANS X9.8, ANS X9.86, ANS X9.87
Biometrics Management and Security	<ul style="list-style-type: none"> • ANS X9.84

In the event that criteria specify that “only” these standards be supported, “these standards” shall be interpreted to refer to those standards, algorithms, and protocols listed above, as well as other relevant standards approved by the following standards organizations: IETF, ANSI X9, ITU-T, ISO, NIST and IEEE.

The system shall use any of the algorithms listed above or those that are supported by any of the standards organizations listed above. If the system

uses any other cryptographic algorithm, then it shall be configurable to also allow the use of an acceptable algorithm in place of the unlisted algorithm. Systems that do not have this flexibility will be disqualified from the BITS validation process.

Appendix B: Glossary of Terms

Definitions provided in this document are provided via references in various books and publications. Several of these references are included at the end of this section.

TERM	DEFINITION
access control	<i>A mechanism for limiting use of some resource to authorized users [1]</i>
account	<i>In terms of a “user account”, an account is an established relationship between a user and a computer, network or information service</i>
active attack	<i>An attack which results in an unauthorized state change, such as the manipulation of files, or the adding of unauthorized files [3]</i>
administrator	<i>Used without pre-qualification, this term indicates any user (or group of users) that could be defined as being a system administrator and/or product administrator, typically having privilege beyond the scope of an end-user. See also “end user”, “user” and “product administrator”.</i>
AIS	<i>Automated Information System - any equipment of an interconnected system or subsystems of equipment that is used in the automatic acquisition, storage, manipulation, control, display, transmission, or reception of data and includes software, firmware, and hardware. [3]</i>
API	<i>Application Programming Interface. Typically provided by a software development toolkit.</i>
asymmetric cryptography	<i>A class of cryptographic algorithms that use separate keys for encryption and decryption. [2]</i>
attack	<i>An attempt to bypass security controls on a computer. The attack may alter, release, or deny data. Whether an attack will succeed depends on the vulnerability of the computer system and the effectiveness of existing countermeasures. [3]</i>
audit	<i>To keep a record of events that might have some security significance, such as when access to resources occurred [1]</i>
authenticate	<i>To determine that something is genuine. To reliably determine the identity of a communicating party [1]</i>
authentication	<i>The process of reliably determining the identity of a communicating party [1]</i>
authenticator	<i>The method, material or credential used to create and/or implement authentication bindings, such as a password, PIN number, token seed, Smart card seed, etc.</i>
authorization	<i>Permission to access a resource [1]</i>
biometric device	<i>A device that authenticates people by measuring some hard-to-forge physical property, like a fingerprint or the strokes and timing of a signature</i>
biometrics	<i>Using physical characteristics of users such as fingerprints and retinal impressions to authenticate users. [2]</i>
buffer overflow	<i>This happens when more data is put into a buffer or holding area, then the buffer can handle. This is due to a mismatch in processing rates between the producing and consuming processes. This can result in system crashes or the creation of a back</i>

TERM	DEFINITION
	<i>door leading to system access. [3]</i>
certificate	<i>A message signed with a public key digital signature stating that a specified public key belongs to someone or something with a specified name [1]</i>
certification authority (CA)	<i>Something trusted to sign certificates [1]</i>
certificate revocation list (CRL)	<i>A list containing names of users and roles that are no longer valid within a public key cryptography system [2]</i>
challenge-response	<i>An authentication mechanism in which the authentication process sends a challenge to a process that requests authentication; the latter is authenticated only if it sends the correct response to the authentication process [2]</i>
clear text	<i>A message or data that is not encrypted</i>
client	<i>Something that accesses a service by communicating with it over a computer network [1]</i>
confidentiality	<i>The property of not being divulged to unauthorized parties [1]</i>
credential	<i>A letter or certificate given to a person to show that he has a right to confidence or to the exercise of a certain position or authority [5]</i>
cryptography	<i>The practice of encoding and decoding data</i>
decrypt	<i>To undo the encryption process [1]</i>
dictionary attack	<i>Typically an “offline attack” or “brute force attack” ... this is the process of “guessing” passwords, based on a set of key words or characters, until a match is made.</i>
digital signature	<i>Digital signatures are used to detect unauthorized modifications to data and to authenticate the identity of the signatory. In addition, the recipient of signed data can use a digital signature in proving to a third party that the signature was in fact generated by the signatory. A digital signature is represented in a computer as a string of binary digits. A digital signature is computed using a set of rules and a set of parameters such that the identity of the signatory and integrity of the data can be verified. An algorithm provides the capability to generate and verify signatures. Signature generation makes use of a private key to generate a digital signature. Signature verification makes use of a public key, which corresponds to, but is not the same as, the private key. Each user possesses a private and public key pair. [6]</i>
DLL	<i>Dynamic Link Library - Software (executable code or data, such as icons or fonts) used by Microsoft's Windows and IBM's OS/2 to provide services (such as a LAN driver or a distributed filing system) to applications. One memory-resident copy of the DLL can be simultaneously shared by all applications.</i>
DNS	<i>Domain Name System - an Internet service that translates domain names into IP addresses</i>
dongle	<i>A device that attaches to a computer to control access to a particular application</i>
end user	<i>Unless otherwise indicated, “end user” is the individual who uses the system after it has been fully developed. The term distinguishes the user for which the product is designed from the developers, installers, and administrators who are making the product available for the end-user. See also “user” and “administrator”.</i>

TERM	DEFINITION
encrypt	<i>To scramble information so that only someone knowing the appropriate secret can obtain the original information (through decryption)</i>
escrow	<i>To hold something in safekeeping. Most uses of the word actually mean keeping the something safe from the owner as opposed to providing any safety for the owner</i>
group	<i>A named collection of users, created for convenience in stating authorization policy</i>
hash	<i>A cryptographic one-way function that takes an arbitrary-sized input and yields a fixed-size output [1]</i>
immutable	<i>Unchangeable [2]</i>
integrity	<i>The quality of being uncorrupted. Message integrity refers to the state of a message not being modified while in transit. File integrity refers to the state of files not being modified while in storage. [2]</i>
integrity checks	<i>(Note: in the context of this document, includes term “data integrity check”): reassuring the recipient of a message that the message has not been altered since it was generated by a legitimate source (based on representing information as numbers and mathematically manipulating those numbers) [1]</i>
interface	<i>In the context of this document, the term “interface” will represent a separate entry point into the system. The term “interface” (i.e. “user interface”) is defined in the context of product administration, indicating the entry point for commands and menu(s) to a system.</i>
key	<i>A quantity used in cryptography to encrypt or decrypt information</i>
key escrow	<i>The system of giving a piece of a key to each of a certain number of trustees such that the key can be recovered with the collaboration of all the trustees [3]</i>
log	<i>To record an action [2]</i>
log file	<i>A file that lists actions that have occurred [2]</i>
MAC	<i>Message Authentication Code – a synonym of message integrity code (MIC) [1]</i>
message digest	<i>An irreversible function that takes an arbitrary sized message and outputs a fixed length quantity. MD2, MD4, and MD5 are message digest algorithms [1]</i>
multifactor	<i>More than two elements or quantities</i>
MIC	<i>Message Integrity Code – a fixed-length quantity generated cryptographically and associated with a message to reassure the recipient that the message is genuine [1]</i>
NIST	<i>National Institute of Standards and Technology</i>
non-repudiation	<i>The property of a scheme in which there is proof of who sent a message that a recipient can show to a third party and the third party can independently verify the source [1]</i>
NTP	<i>Network Time Protocol</i>
OCSP	<i>Online Certificate Status Protocol</i>
offline attack	<i>An attack performed while offline to the system being attacked (see also “dictionary attack”)</i>
one-time passwords	<i>Passwords that can only be used one time [2]</i>

TERM	DEFINITION
operator	<i>In the context of this document, “operator” maintains similar relationships and functions as “administrator” (see above), given different and/or additional privileges than a typical “end user” of a system.</i>
orthogonal	<i>Having to do with right angles; rectangular [5]</i>
passive attack	<i>Attack which does not result in an unauthorized state change, such as an attack that only monitors and/or records data [3]</i>
password	<i>A supposedly secret string used to prove one’s identity [1]</i>
PIN	<i>Personal Identification Number – a short sequence of digits used as a password [1]</i>
PKCS	<i>Public Key Cryptography Standards (PKCS) ... a set of standards, first introduced in 1991 by RSA Data Security, Inc., for implementing public key cryptographic algorithms and incorporating them in to applications [2]</i>
plaintext	<i>Unencrypted data [3]</i>
pre-authentication	<i>A protocol for proving you know your password before you are allowed access to a high quality secret encrypted with that password [1].</i>
private key	<i>The quantity in public key cryptography that must be kept secret [1]</i>
privileged user	<i>A user of a computer or system who is authorized to bypass normal access control mechanisms, usually to be able to perform system management functions [1]</i>
product administrator	<i>In the scope of this document, this role is associated with a user having higher privilege at the product’s configuration level. This user (role) may be the same as the system administrator, but could also be different.</i>
protected path	<i>A mechanism that guarantees a mutually authenticated channel [4]</i>
public key	<i>The quantity in public key cryptography that is safely divulged to as large an extent as is necessary or convenient [1]</i>
public key cryptography	<i>A cryptographic system where encryption and decryption are performed using different keys – see Asymmetric key cryptography [2]</i>
relying party	<i>In the scope of this document, “relying party” is typically associated with the use of “system” (see below). It is associated frequently with the extent of which a criteria is in scope as in references to an application or component maintaining reliance on another element to support said criteria.</i>
replaying	<i>Storing and retransmitting messages. The word is usually used when implying that the entity doing the reply of messages is mounting some sort of security attack</i>
repudiation	<i>Denying that you did something or made some statement [1]</i>
revoke	<i>To withdraw, repeal, rescind, cancel, or annul [5]</i>
role	<i>A function or office assumed by someone [5]</i>
security domains	<i>The sets of objects that a subject has the ability to access [3]</i>
security features	<i>The security-relevant functions, mechanisms, and characteristics of AIS hardware and software [3]</i>
server	<i>Some resource available on the network to provide some service such as name lookup, file storage, or printing [1]</i>

TERM	DEFINITION
sign	<i>To use your private key to generate a digital signature as a means of proving you generated, or approve of, some message</i>
signature	<i>A quantity associated with a message which only someone with knowledge of your private key could have generated, but which can verified through knowledge of your public key [1]</i>
spoof	<i>To convince someone that you are some entity X when you are not X, without X's permission [1]</i>
strong authentication	<i>Authentication performed in such a way that it cannot easily be performed. Examples of strong authentication include one-time passwords, challenge-response mechanisms, and cryptographic authentication [2]</i>
symmetric key cryptography	<i>A class of cryptographic algorithms in which the same key is used for encryption and decryption. Examples of symmetric key algorithms include DES, IDEA, RC2, and RC4 [2]</i>
system	<i>The term "system" is used to imply the totality of the product and the mediation device (if any) that need to be tested.</i>
system administrator	<i>In the scope of this document, an individual (user) having higher privilege at the operating system level.</i>
system restart	<i>To restart a system. May also be referenced as a "warm boot", whereas the system is restarted from an operational state. Additionally, may also be referenced as a "cold boot", whereas the system is powered off and then on again</i>
token device	<i>A credit-card sized device that generates authentication tokens, such as one-time passwords [2]</i>
two-factor authentication	<i>A process in which two pieces of information are required to prove one's identity (such as a password and a smart card) [2]</i>
weak authentication	<i>Typically, this implies the conventional use of passwords</i>
user	<i>Used without pre-qualification, this term indicates any and all users, such as end-user, product user-ID or system user.</i>
user-ID	<i>A number or name which is unique to a particular user of a computer or group of computers which share user information. The operating system uses the uid to represent the user in its data structures, e.g. the owner of a file or process, the person attempting to access a system resource etc.</i>
X.509	<i>A CCITT standard for security services within the X.500 directory services framework. The X.509 encoding of public key certificates has been widely adopted; the other protocol elements of X.509 have not [1]</i>

Glossary Items are based on the following references:

- ◆ [1] Kaufman, C., Perlman R. and Speciner M., *Network Security: Private Communication in a Public World*, Prentice Hall, New Jersey, 1995

- ◆ [2] Bernstein, T., Bhimani A., Schultz E., and Siegel C., *Internet Security for Business*, John Wiley & Sons, Inc., New York, 1996
- ◆ [3] NSA Glossary of Terms used in Security and Intrusion Detection
- ◆ [4] Loscocco Peter A., Smalley Stephen D., Muckelbauer Patrick A., Taylor Ruth C., Turner S. Jeff, Farrell John F., *The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments*, 1998
- ◆ [5] Guralnik, David Bernard (editor), Webster's New World dictionary of the American Language, Prentice Hall Press, 1986
- ◆ [6] FIPS PUB 186-2, *DIGITAL SIGNATURE STANDARD (DSS)*, 27 January 2000