U.S. DEPARTMENT OF HEALTH AND HUMAN SERVICES

Office of Information Services
Chief Information Officer

CMS Web-Enabled Application Architecture

Version 1.1

March 2004
(Updated: June 2005)

CMS
CENTERS for MEDICARE & MEDICAID SERVICES

Document Number:
CMS-CIO-STD-INT03

TABLE OF CONTENTS

FIGURES AND TABLES

1. FOREWORD

This document provides an overview of the Centers for Medicare & Medicaid Services (CMS) Web-Enabled Application Architecture. The CMS Web-Enabled Application Architecture addresses CMS applications from a common set of security services perspective. This architecture identifies a common set of security services that addresses the needs of the CMS Enterprise, and establishes the associated application level security controls required for all CMS applications.

The Office of Information Services' (OIS) Deputy Director for Technology/ Chief Technology Officer leads the development of this architecture, including the overall Internet architecture, with the support of all OIS components to include contributions from other CMS Centers and Offices. It is applicable and serves as the blue print for the implementation for all in-house and contractor applications in support of CMS business operations.

/s/ 7/8/05

D. Dean Mesterharm *Date*
Director, Office of Information Services
CMS Chief Information Officer

/s/ 7/8/05

Wallace K. Fung *Date*
Deputy Director, Office of Information Services
CMS Chief Technology Officer

2. EXECUTIVE SUMMARY

The Centers for Medicare & Medicaid Services (CMS) is in the process of modernizing its information technology (IT) capabilities. The CMS enterprise architecture will be reengineered into a three-tier environment comprised of three zones: Presentation, Application, and Data. CMS applications will provide web-based access to all their users. Users will employ web browsers to access web servers in the Presentation Zone. The web servers will be front-ends for CMS enterprise applications running in the Application Zone. And, all application data will be stored in the Data Zone. This architecture will be supported by a common set of enterprise security services integrated into the three zones.

This document outlines application security services components and establishes specific sets of security controls that applications will employ to provide security uniformity and consistency for all CMS systems and information. This document supersedes all previous draft versions.

3. BACKGROUND

One of CMS' Strategic Goals is to "streamline our environment so that existing and new systems can work more effectively by sharing information, and so that CMS can be more responsive to the demands of changing business needs and the promises of emerging technology. We plan to make our data more readily accessible to our beneficiaries, partners, and stakeholders in a secure, efficient, and carefully planned manner."[1]

In striving to meet these goals, CMS is in the process of migrating to a web-based environment in which services will be provided by web applications. To support this web-based environment, the current architecture is being engineered into three zones - the Presentation Zone, Application Zone, and Data Zone. This architecture provides an increased degree of security since its multiple zones isolate protected CMS information systems and data.

As a result of the large volume of outsourcing required in recent years to automate CMS services, there has not been a unified approach to enterprise security. Individual projects and applications have employed a multitude of varying security methods and mechanisms for the protection of confidential healthcare information. Unfortunately, the coordination between these efforts has not always been the best, and as a result, CMS now has applications within our enterprise, which are architecturally dissimilar to each other. Furthermore, few of these applications have components that are portable across the different types of hardware operating environments within CMS.

This document seeks to establish a common set of security services and required security controls that shall be employed by all application implementation efforts to help ensure:

- Organization-wide uniformity in compliance with federal regulations;
- A reasonable level of assurance to end-users of system trustworthiness; and
- Uniform levels of confidentiality, integrity, and availability for all healthcare information created, received, processed, stored, or transmitted by CMS.

The security controls described within this document have been adapted from the National Institute of Standards and Technology (NIST) Special Publication (SP) 800-37, *Federal Guidelines for the Security Certification and Accreditation of IT Systems;* SP 800-53, *Recommended Security Controls for Federal Information Systems;* and SP 800-63, *Recommendation for Electronic Authentication,* all currently in draft form. When these documents are formally released, the *CMS Web-Enabled Application Architecture* will be reviewed and modified accordingly.

[1] Taken directly from the CMS Strategic Goals and Objectives statement

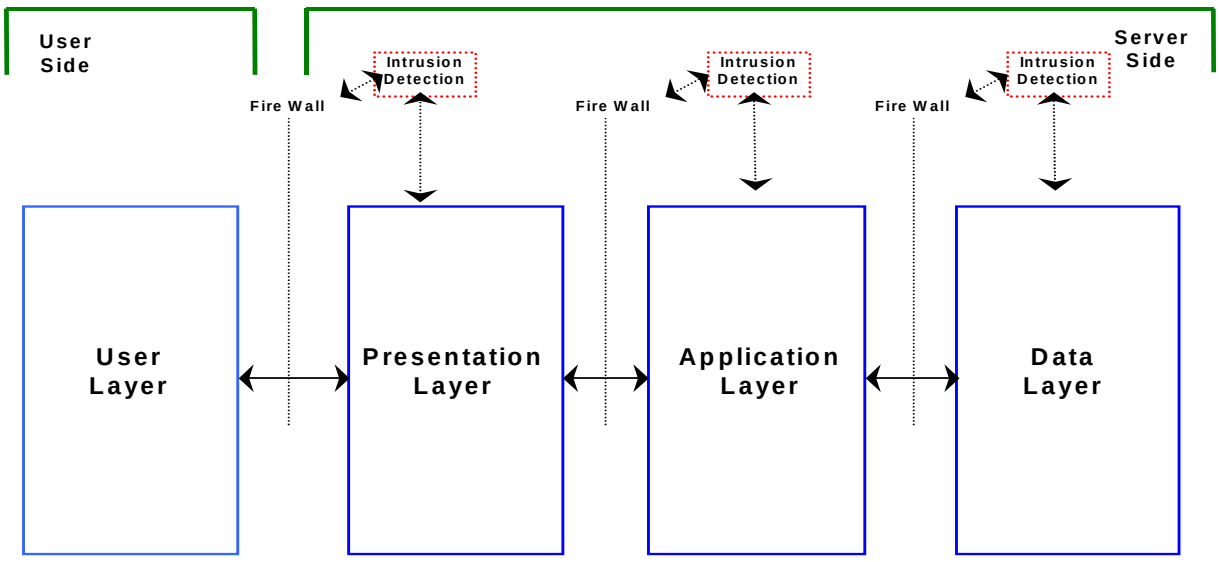
4. Application Architecture, Security View

The **Application Architecture, Security View** closely resembles the three-tier architecture with one difference in that it shows the User Layer, or user interface. Each of these application layers is presented to show the major application technical support components and their relationships with each other.

Online applications have many common characteristics. With few exceptions, applications comprise four major Layers: a User Layer, a Presentation Layer, an Application Layer, and a Data Layer.

Batch applications may **also** possess these same four major layers. The Data Layer, the Application Layer and the Presentation Layer (if well implemented) should have a similar look and feel to the corresponding Layers for an online application. Instead of the web browser interfaces, the batch application's User Layer would consist of bulk files for input, and bulk files (or printed reports) as output. Batch applications within the User Layer may be an important and continuing feature of this architecture, but will be discussed in detail in other, more application-specific documents wherever applicable. The remainder of this document describes only the Online User Layer in relation to the other three layers.

Figure 1: Application Architecture, Security View



In Figure 1, the **Application Architecture, Security View** is shown as four layers in two groups: a "user-side" layer and three "server-side" layers:

4.A. User Layer

The User Layer consists of all the services to be directly utilized by the end user when the system is accessed. Users utilize a web browser that interacts with Server Side Components in order to carry out the tasks that the end user desires.

4.B. Presentation Layer

The Presentation Layer is the first of the three Server Side Components that control how the application looks and to some degree how it interacts with the user. Essentially, all of the screens, forms, Graphic User Interfaces (GUIs), etc. will be composed within the Presentation Layer and passed to the User Layer for viewing and possible response. This Layer separates the User from the Application and Data resources and plays a key role in the overall security scheme. Information from the screens, forms, GUIs, etc. that is furnished by the user will be passed to, received, and reformatted by the Presentation Layer and then passed to the Application Layer for processing.

4.C. Application Layer

The Application Layer is the second of the three Server Side Components that performs the application-specific functions not directly related to the user interface. This includes data validation, execution of business rules, calculations, manipulation of data, control of the environment, etc. When the Application Layer requires information or actions from the Data Layer, it is requested from the Data Layer by way of a messaging facility. The Application Layer then processes the responses by way of a messaging facility from the Data Layer. Replies from the Application Layer are then returned to the Presentation Layer, subsequently reformatted by the Presentation Layer and then returned to the User Layer for viewing.

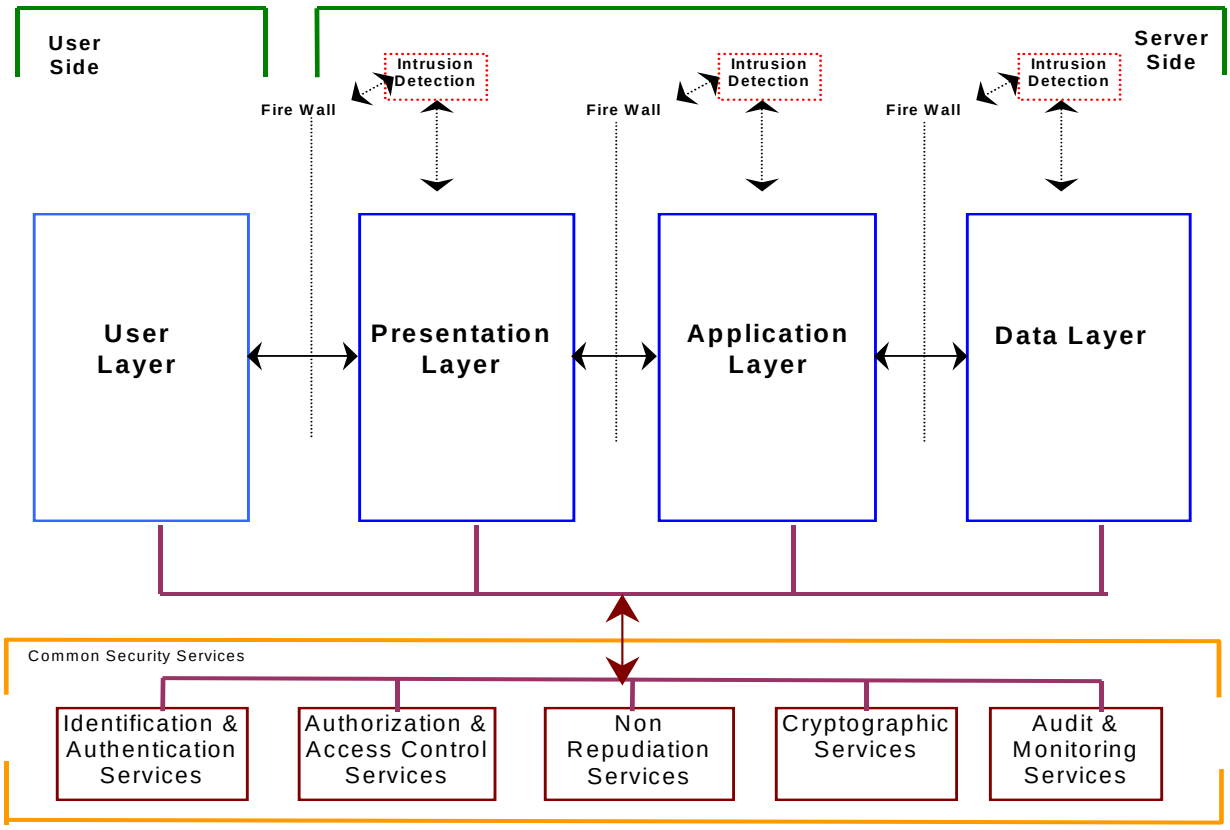
4.D. Data Layer

The Data Layer is the third component on the Server Side and represents the "back-end" data resources for the application. The most prominent data resource would be the Relational Database Management System (RDBMS). Any required information in the Data Layer would be provided by the Application Layer. The Data Layer will interface only with the Application Layer through the messaging facility, separating it completely from the Presentation Layer for security purposes. Any Data sources external to the application (i.e., data that resides in other RDBMSs) would be made available to the Application Layer by way of the messaging facility from the Data Layer.

5. Application Security Architecture

The **Application Security Architecture, High Level View** is presented to expand and further develop the components (or layers) discussed in the Application Architecture, Security View.

Figure 2: Application Security Architecture, High Level View



As shown in Figure 2, there will be firewalls between each of the major layers, giving CMS the ability to exercise more granular control over each layer. For specific security requirements for firewalls, please refer to the *CMS Internet Architecture (Including Minimum Platform Security Requirements)* dated July, 2003.

Intrusion Detection Hardware/Software components are to be deployed at each layer of the architecture and offer the opportunity to detect and combat a break-through at the outermost layers before the inner layers are compromised. Each physical computer platform will have its own Operating System level security package implemented. The same will be true for the DBMS and Messaging Software. Security in these environments is more extensive than what is discussed in this document. For the messaging component, please refer to the *CMS Enterprise Messaging Infrastructure Including Architecture, Standards, and Implementation Requirements*

dated December, 2003. The *CMS Intrusion Detection System Internet Architecture and Design* and the *CMS Data & Database Architecture* documents are currently being developed.

Figure 2 also presents Common Security Services families, which include:

- Identification and Authentication Services
- Authorization and Logical Access Control Services
- Non-Repudiation Services
- Cryptographic Services
- Audit & Monitoring Services

Each of the Common Security Services families will provide the core security mechanisms that applications will employ. Only the Common Security Services relevant to applications are depicted in Figure 2 and will be referred to as Application Security Services in the remainder of this document. The other Common Security Services applicable to Infrastructure Security Services are not depicted here. The *CMS Information Security Acceptable Risk Safeguards (ARS), Version 1.2*, dated October 25, 2004, provides Infrastructure Security Standards by system security levels.

Application Security Services are those security services that are typically visible to the application developer and most effective when transparent to the end user. Applications shall make use of the available Application Security Services and should understand security implicitly. In cases where commercially available applications are used, applications should be configured with advanced security controls within the application, should incorporate the available Common Security Services whenever possible, and comply with CMS enterprise security policies for access control and handling.

5.A. Application Security Controls

This section outlines the required security controls within each of the Security Services families that applications shall employ in order to minimize information security risk. These controls shall be applied to the particular architectures upon which an application is implemented, and as such, are not specific to the level of configuration settings or application standards. Taken together these security controls are to be used for specifying application-level security requirements for IT products and systems during acquisition, design, development or implementation.

Deploying a secure application entails implementing technical application security controls at the application layer, but requires adequate system and network security controls. Only a combination of controls at each of these levels will yield an application that is sufficiently secure.

As is often the case, multiple applications may share platforms, databases or other system resources and certainly share network resources. This resource sharing, combined with the increasing utilization of common security services for provisioning of technical application security controls across a broad range of subscribing applications, often means that application developers do not have direct control over the manner in which all technical application security controls are deployed.

*However, it remains the responsibility of the application developer to select, implement, and document the nature and type of technical application security controls employed in a particular application design. These controls shall conform to the requirements of a specific system security level (refer to the **CMS Information Security Levels** and **Acceptable Risk Safeguards**) and be such that the implementation decisions conform to information security risk management goals. CMS Management shall also approve all the controls implemented for each system development and deployment effort.*

The application security controls are grouped into families based on the services they provide. The families of technical security controls are:

- Identification & Authentication
- Authorization and Logical Access Control
- Non-Repudiation
- Cryptographic
- Audit & Monitoring

A unique naming convention for security controls is used to help describe the family from which the control component is selected and the number of the control component within the family. For example, a security control component element identified as IA-4.3 indicates that a control component is:

(IA) From the Identification and Authentication family,
(4) A Specification of Passwords sub-group requirement, and
(3) The third in that sub-group within that family.

The key words "Shall", "Shall not", "Should", "Should not", are used throughout the following application security controls to signify functional requirements or system behaviors. They are to be interpreted as described below:

- **Shall**: The functional requirement is an absolute requirement of the specification.
- **Shall not**: The functional requirement is an absolute prohibition of the specification.
- **Should**: Valid reasons may exist in particular circumstances to not strictly comply with a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
- **Should not**: Valid reasons may exist in particular circumstances when the behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

5.A.1. Identification and Authentication Services (Family – IA)

A secure system is required to ensure the unambiguous identification of subjects or entities and the correct association of security attributes (e.g. identity, groups, roles, security, or integrity levels) with these subjects. Correct identification and authentication of subjects is critical to enforcement of security policies. The terms identification and authentication are often used interchangeably, but should not be. *Identification* is the process of asserting the identity of an entity (human or computer) in some security domain. A typical example of a subject is a person, identified by his or her email address in a particular Internet DNS domain process.

The use of identification alone is, however, usually not sufficient. It does not provide assurance that a given subject is a legitimate user or system. To solve this problem, the authentication process comes into play. *Authentication* is the process of verifying the claimed identity of a subject or entity based upon protocols.

For those applications requiring identification and authentication, the following component elements constitute the CMS Application Identification and Authentication security control requirements:

IA –1 Identification– subjects are required to assert an identity before any other enterprise application actions are permitted.

IA-1.1 The application shall enforce identity assertion through a unique identifier.

IA-2 Subject Security Attribute Definition– all authorized entities shall have a set of security attributes, other than the entity's identity, that is used to enforce enterprise application security policy.

IA-2.1 The following list of security attributes shall be maintained for human entities:

- Secret password (Optional: token, certificates, etc.)

IA-2.2 The following list of security attributes shall be maintained for computer entities:

- Secret password (Optional: token, certificates, etc.)

IA-3 User-Subject Binding – an authorized administrator shall maintain the binding between a user-subject and the defined security attributes. System developers shall not be permitted to assign security attributes.

IA-3.1 The application shall only provide interfaces to change user-subject bindings only to explicitly identified administrators.

IA-4 Specification of Passwords – passwords shall be verified against defined quality metrics. Passwords that do not meet quality metrics according to organizational policies and standards shall be rejected.

IA-4.1 The application shall utilize passwords that are at least eight characters in length and contain a mixture of alphabetic and numeric symbols.

IA-4.2 The application shall enforce password expiry timeframes. (i.e. passwords shall be changed every 60 days and ensure discontinued access in case of password/secret expiry.)

IA-4.3 The application shall enforce password quality metrics and ensure discontinued access in case of quality metric failure.

IA-4.4 The application shall be able to generate passwords that meet the quality metric.

IA-5 Authentication – defines the protocol that determines if claimed security attributes validate a subject or entities identity. Types of authentication that may be required include:

IA-5.1 Applications shall enforce entity authentication by use of a password before system access is allowed.

IA-5.2 The application shall utilize multiple authentication mechanisms.

IA-5.3 The application shall force re-authentication after a period of specified time has elapsed since the application has been utilized.

IA-5.4 The application shall force re-authentication after the user entity has attempted to execute an application or portion of an application with a higher assurance level.

IA-5.5 The application shall force re-authentication after the user entity has entered a portion of the application that is out of sequence of the normal use of the application. While this may often be avoided through robust application design that only allows users to execute actions associated with their role, this control is intended specifically to reduce the risks associated with user session hijacking and terminal takeover.

IA-6 Authentication Failure Handling – defines the requirements related to the number of unsuccessful authentication attempts in a given time period after which the application shall be able to terminate the session establishment process. This section also defines enforcement actions in case of authentication attempt failures. This section contains control standards that are often mutually exclusive, and as such, utilizing a higher level of control may necessarily dictate that the lower level controls may have to be removed from the application. It should be noted that authentication failure might be handled by the application directly, or by a network-based authentication service. In either case, it is the application owner's responsibility to ensure that an appropriate authentication failure control mechanisms are accounted for in their application designs.

IA-6.1 Applications shall not provide feedback upon authentication failure indicating the validity of either identification or authentication information.

IA-6.2 The application shall be able to detect three unsuccessful authentication attempts in a 30 minute time period. Furthermore, such a series of failed authentication attempts shall cause access to be denied to the entity until an authorized administrator performs an application access reset.

IA-6.3 The application shall be able to detect three application access resets in a one week time period. Furthermore, such a series of application access resets shall cause the application to alert the authorized administrator to notify the entity's administrative supervisor.

5.A.2. Authorization and Logical Access Control Services (Family - AC)

Once a user has been identified and authenticated, it still remains to be determined what actions a given user is allowed to perform and what information that user is allowed to access. The process of making this determination is called *authorization*. *Logical Access Control* refers to security policies and functions related to protecting user data from *unauthorized* users.

The following component elements constitute the CMS Authorization and Logical Access Control security control requirements:

AC-1 Access Control Polices - Role Based Access Control (RBAC) – access decisions are based upon the concept of roles and responsibilities. The access control framework should provide the ability to determine which entity shall perform what actions, when, from where, in what order, and in some cases, under what relational circumstances.

AC-1.1 The application shall only allow user entities access to components or application systems, functions, and data based on their authorization level. RBAC should be utilized to prevent unauthorized access whenever possible to allow for ease of system security administration. Additionally, RBAC should be utilized to allow for the provisioning of user access privileges in a hierarchical manner (i.e., only users in the Jacksonville office who belong to the Customer Support group may access the application.)

AC-1.2 The application should utilize the user entity's specific security attributes to determine access policies.

AC-2 Access Control Functions – these controls provide a mechanism that mediates access control based on security attributes associated with subjects and objects. Typical attributes include identity, timestamp, location, and groups.

AC-2.1 The application shall utilize the identity of the user entity to determine access.

AC-2.2 The application should utilize the identity of the user entity and timestamp to determine access.

5.A.3 Non-Repudiation Services (including Digital Signatures) (Family – NR)

No security control requirements have been developed for non-repudiation services at this time. Future security control requirements are planned and will be dealt with in greater depth in future releases of this document.

5.A.4 Cryptographic Services (Family – CS)

Cryptographic support is used to meet many operational security objectives including the protection of data communications and data storage. Other operational objectives that require cryptographic support include digital signature generation and verification, checksum generation and integrity verification, secure hash (message digest) computation, data encryption and decryption, cryptographic key encryption and decryption, revocation list, certification arbitration, key escrow, key agreement, and random number generation.

The following component elements constitute the CMS Cryptographic security control requirements:

CS-1 Key Management - Cryptographic keys shall be managed throughout their lifetime. The typical cryptographic key lifecycle includes: generation, distribution, storage, access (e.g., backup, escrow, archive, recovery) and destruction. The key management strategy implemented by an application shall, in large measure, determine the assurance level achievable by the application. The following components represent the minimum standards that a key management strategy shall meet.

CS-1.1 Key Generation – this component specifies the cryptographic key size and algorithm to be used to generate cryptographic keys. The minimum standards are:

- The AES cryptographic algorithm is the accepted key generation algorithm.
- A key length of 128-bit or greater shall be used for domestic data communication within the United States.
- Key lengths of 1024-bits and greater shall be used for non-repudiation services.

CS-1.2 Key Distribution – this component specifies the method used to distribute cryptographic keys. Applications are required to implement the RSA Authenticated Key Exchange Algorithm distribution method.

CS-1.3 Key Storage - all applications using cryptographic keys for secure communications (messaging) and non-repudiation services shall specify the process for storing keys. These processes shall address administrative access to the host and host file system and the means for auditing and monitoring key integrity while in storage.

CS-1.4 Key Access – all applications using cryptographic keys for secure communications (messaging) and non-repudiation services shall specify a key backup, archival, escrow and recovery process. No minimum standards have been set at this time.

CS-1.5 Key Destruction - all applications using cryptographic keys for secure communications (messaging) and non-repudiation services shall specify a key destruction process. No minimum standards have been set at this time.

CS-2 Key Operations - A cryptographic operation (e.g. digital signature generation, secure hash computation, message encryption/decryption) shall have a cryptographic mode associated with it. Cipher Block Chaining (CBC) mode is the minimum standard mode for cryptographic operation.

CS-3 Cryptographic Operations – Cryptography, based on the cryptography standards outlined herein, should be utilized to protect data during communications and storage at various points during its use by the application. Specifically with respect to encryption key lengths, as export standards and key lengths mature, these standards may need to be modified continually.

CS-3.1 The application shall encrypt data communicated between the Presentation Layer and the User Layer.

CS-3.2 The application shall be able to encrypt data communicated between the Presentation Layer and the User Layer using a minimum of 128-bit encryption.

CS-3.3 The application shall be able to encrypt data communicated between all application component layers using a minimum of 128-bit encryption.

CS-3.4 The application shall utilize digital signatures to enable non-repudiation and additional audit capabilities.

5.A.5 Audit and Monitoring Services (Family – AM)

Security auditing involves recognizing, recording, storing, and analyzing information related to security events or actions. A properly configured audit trail system shall, at a minimum, answer the questions, “Who did what, when, where, and how?”

Intrusion Detection Systems (network/ host /application based) and firewalls can communicate whether a problem has occurred, but determining the full extent of that intrusion requires an in-depth analysis of detailed records that form an audit trail. An audit trail system chronologically records user, system, application, and network activities to an extent sufficient to enable the reconstruction of a sequence of events.

While a properly configured audit trail system will record transaction data, it cannot by itself prove the parties involved, authorization level of participants, improper disclosure of information, and integrity of the transaction.

Applications shall be developed to be able to identify the types of interactions occurring in real-time, or that have occurred historically, between users and objects. Such real-time monitoring or audit logging and analysis shall be built intrinsically into the application, or be provided via a suitable messaging system to a common monitoring and audit log analysis service.

Real-time monitoring is applied to communication or messaging between n-tier application layers through a process of rules induction. Typical rules engines employ inductive logic to message syntax and semantics; to conformity of communications with application-layer protocols or message grammar; and to the behavior of the communicating entities (human, application software or an underlying computer system.) Providing accountability through the use of both real-time/near-real-time monitoring and historical auditing of application controls is a critical factor in assuring application security goals and system trustworthiness.

The following component elements constitute the CMS Audit and Monitoring security control requirements:

AM-1 Audit Data Generation – specifies the set of auditable events.

AM-1.1 The application should hierarchically categorize significant transactions such that information on transaction execution may be collected and organized.

AM-1.2 The application shall log and track all application-specific authentication events, to include both successful and failed logons.

AM-1.3 The application shall log and track all application-specific actions of users with special-access privileges (i.e.; administrative privileges), password changes, user account creations and removals.

AM-1.4 The application should provide an administrative interface through which the logging and auditing of significant events may be enabled by selecting the hierarchical level at which logging should occur (i.e. the administrator assigns a severity level to certain events, then log all events with severity less than level 2.)

AM-1.5 The application should provide an administrative interface through which the hierarchical categorization of transactions may be changed (i.e. record changes are no longer severity level 1, they are severity level 2.)

AM-1.6 The application should provide an administrative interface through which audit logs shall be generated and maintained. At a minimum, the interface shall provide the administrator the ability to modify log file storage location(s), as well as select and modify the configuration of the log file (i.e.; log size limitations, retention policies, type of information recorded in the log.)

AM-2 Audit Analysis & Logging – specifies a system for detecting possible or real security violations.

AM-2.1 The application should be able to generate an event based on a fixed rule regarding occurrence of a transaction of a defined severity.

AM-2.2 The application should be able to generate an event based on data being submitted that is not within the bounds of acceptable values or that contains certain keywords (i.e. SQL piggybacking.)

AM-2.3 The application should be able to generate an event based on deviation from a pre-defined order of operations (i.e. entity attempts a transaction that does not follow the previous transaction normally.)

AM-3 Automatic Response – describes requirements for handling audit events.

AM-3.1 The application should be able to record specified events that are deemed a threat to application security. The application shall be able to collect event data such as time of occurrence, severity, and entity responsible for the event.

AM-3.2 The application should generate an alert using at least two of the following mechanisms upon detection of an event that is deemed to indicate a threat to application security.

- SNMP Alert or 'trap' - report to Host Intrusion Detection System (HIDS)
- SMTP message - report to HIDS
- Syslog or other logfile entry creation
- Other protocol/mechanism

AM-3.3 The application should immediately disallow application access to the offending entity and cancel transactions being executed by that entity upon detection of an event that is deemed a threat to application security.

AM-4 Audit Review – provides guidelines for audit data storage, recovery and inspection.

AM-4.1 The application shall store audit data within a local file that is secured such that only the assigned security staff may access it.

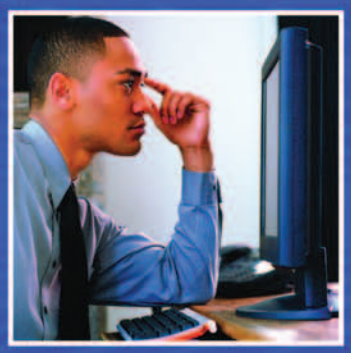
AM-4.2 The application shall send data to a centralized audit store and maintain a local copy of a minimum of two days worth of audit data if connectivity to the centralized store is compromised.

5.B Application Security Requirements

The following table details the applicability of security control requirements based upon the determined sensitivity and criticality level of the system. The requirements are broken down into the Common Security Services groups for organizational purposes. The intent of this listing is to provide a high-level overview of the required security controls that applications shall satisfy at various levels of security according to criteria set forth in *the CMS Information Security Levels* document, which is either: "low," "moderate," or "high." Each of these values dictates a specific combination of application level security controls that shall be designed into the system. However, the listing of controls does not necessarily indicate increasing levels of application security.

Table 1: Cross-Reference of System Security Levels and Required Security Controls

Common Security Services Family	Required Security Control Components	Required Control by System Security Level		
		Low	Moderate	High
Identification and Authentication (IA)	IA-1.1	X	X	X
	IA-2.1	X	X	X
	IA-2.2	X	X	X
	IA-3.1	X	X	X
	IA-4.1	X	X	X
	IA-4.2	X	X	X
	IA-4.3	X	X	X
	IA-4.4			X
	IA-5.1	X	X	X
	IA-5.2			X
	IA-5.3	X	X	X
	IA-5.4			X
	IA-5.5			X
	IA-6.1	X	X	X
	IA-6.2	X	X	X
	IA-6.3	X		
Authorization and Logical Access Control (AC)	AC-1.1	X	X	X
	AC-1.2	X	X	X
	AC-2.1	X	X	X
	AC-2.2			X
Non-Repudiation (NR)	No Specification at this time			
Cryptographic Controls	CS-1.1	X	X	X
	CS-1.2		X	X
	CS-1.3		X	X
	CS-3.1	X	X	X
	CS-3.2		X	X
	CS-3.3			X
	CS-3.4			X
Audit and Monitoring (AM)	AM-1.1		X	X
	AM-1.2	X	X	X
	AM-1.3	X	X	X
	AM-1.4		X	X
	AM-1.5		X	X
	AM-1.6	X	X	X
	AM-2.1		X	X
	AM-2.2	X	X	X
	AM-2.3			X
	AM-3.1		X	X
	AM-3.2		X	X
	AM-3.3			X
	AM-4.1	X	X	X
	AM-4.2			X



OIS Office of Information Services

Centers for Medicare & Medicaid Services
7500 Security Boulevard
Baltimore, MD 21244-1850

www.cms.hhs.gov
www.medicare.gov