Network Working Group
Request for Comments: 3881
Category: Informational

G. Marshall
Siemens

September 2004

Security Audit and Access Accountability Message XML Data Definitions for Healthcare Applications

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

IESG Note

This RFC is not a candidate for any level of Internet Standard. The IETF disclaims any knowledge of the fitness of this RFC for any purpose, and notes that it has not had IETF review. The RFC Editor has chosen to publish this document at its discretion.

Abstract

This document defines the format of data to be collected and minimum set of attributes that need to be captured for security auditing in healthcare application systems. The format is defined as an XML schema, which is intended as a reference for healthcare standards developers and application designers. It consolidates several previous documents on security auditing of healthcare data.

Table of Contents

1. Purpose

To help assure healthcare privacy and security in automated systems, usage data needs to be collected. This data will be reviewed by administrative staff to verify that healthcare data is being used in accordance with the healthcare provider's data security requirements and to establish accountability for data use. This data collection and review process is called security auditing.

This document defines the format of the data to be collected and minimum set of attributes that need to be captured by healthcare application systems for subsequent use by an automation-assisted review application. The data includes records of who accessed healthcare data, when, for what action, from where, and which

patients' records were involved. The data definition is an XML schema to be used as a reference by healthcare standards developers and application designers.

This document consolidates previously disjointed viewpoints of security auditing from Health Level 7 (HL7) [HL7SASIG], Digital Imaging and Communications in Medicine (DICOM) Working Group 14, Integrating the Healthcare Enterprise (IHE) [IHETF-3], the ASTM International Healthcare Informatics Technical Committee (ASTM E31) [E2147], and the Joint NEMA/COCIR/JIRA Security and Privacy Committee [NEMASPC]. It is intended as a reference for these groups and other healthcare standards developers.

The purposes the document fulfills are to:

1) Define data to be communicated for evidence of compliance with, or violations of, a healthcare enterprise's security and privacy policies and objectives.

   This document defines the audit message format and content for healthcare application systems. The focus of auditing is to retrospectively detect and report security/privacy breaches. This includes capturing data that supports individual accountability for patient record creation, access, updates, and deletions.

   This document does not define healthcare security and privacy policies or objectives. It also does not include real-time access alarm actions since there is a perception in the healthcare community that security measures that inhibit access may also inhibit effective patient care, under some circumstances.

2) Depict the data that would potentially reside in a common audit engine or database.

   Privacy and security audit data is to be collected on each hardware system, and there are likely to be separate local data stores for system-level and application-level audits. Collating these records and providing a common view - transcending hardware system boundaries - is seen as necessary for cost-effective security and privacy policy administration.

   The data definitions in this document support such a collation, but the technical implementation alternatives are not covered in this document.

3) Depict data that allows useful queries against audited events.

   Audit data, in its raw form, reflects a sequential view of system activity. Useful inquiries for security and privacy administration need workflow, business process, organizational, role, and person-oriented views. Data definitions in this document anticipate and support creating those views and queries, but do not define them.

4) Provide a common reference standard for healthcare IT standards development organizations.

   By specifying an XML schema, this document anticipates extensions to the base schema to meet requirements of healthcare standards bodies and application developers.

2. Scope

2.1. Data Collection

This document specifies audit data to be collected and communicated from automated systems. It does not include non-automated processes.

Data for events in the above categories may be selectively collected, based on healthcare organization policy. This document does not specify any baseline or minimal policies.

For each audited event, this document specifies the minimal data requirements plus optional data for the following event categories:

1) Security administrative events - establishing and maintaining security policy definitions, secured object definitions, role definitions, user definitions, and the relationships among them. In general, these events are specific to the administrative applications.

2) Audit access events - reflecting special protections implemented for the audit trail itself.

3) Security-mediated events - recording entity identification and authentication, data access, function access, nonrepudiation, cryptographic operations, and data import/export for messages and reports. In general, these events are generic to all protected resources, without regard to the application data content.

4) Patient care data events - documenting what was done, by whom, using which resources, from what access points, and to whose medical data. In general, these audits are application-specific since they require knowledge of the application data content.

Security subsystems found in most system infrastructures include a capability to capture system-level security relevant events like log-on and security object accesses. This document does not preclude such functions being enabled to record and supply the data defined in this document, but transformation of the collected data to the common XML schema definition may be necessary to support requirements consolidated auditing views.

Application-level events, such as patient record access, are not captured by system-level security audits. The defined data support applications' record access auditing for healthcare institutional security and privacy assurance plus related policy administration functions.

System-local data definitions for collection and storage of audit data, prior to transformation to a common schema and transmission to a common repository, are not included in this document.

2.2. Anticipated Data End-uses

This document anticipates, but does not define, end-uses for the data collected.

The typical healthcare IT environment contains many systems from various vendors and developers who have not implemented common or interoperable security administrative functions. This document anticipates a requirement to transmit data from several unrelated systems to a common repository. It also anticipates the aggregated data which may then be queried and viewed in a variety of ways.

There are distinctions of detail granularity, specificity, and frequency between audit data required for surveillance versus forensic purposes. While some surveillance data may be useful for forensics, the scope of this document is limited to surveillance.

This document does not address access real-time policy violation alarm actions. There is a perception in the healthcare community that security measures which inhibit access may also inhibit effective patient care, under some circumstances.

This document does not define any data for patient care consents or patients' permissions for data disclosure. It is conceivable that the proposed audit data could be input to such applications, however, assuming strict access controls for audit data have been established.

This document does not define system-specific or application-specific data that may be collected and reported in addition to the defined elements. For example, it is conceivable that audit mechanisms may be useful for tracking financial or payroll transactions. At the same time, this document does not preclude extending the XML schema to incorporate additional data.

There is a potential requirement for a set of administrative messages to be sent from a central source to each participating system to uniformly specify, control, enable, or disable audit data collection. Such messages are not included in this document.

2.3. Conformance

This document does not include any definitions of conformance practices. Instead, it anticipates that standards development organizations that reference this document may specify their own conformance requirements.

3. Goals

3.1. Effective Data Gathering

The process of assuring that security policies are implemented correctly is essential to information security administration. It is a set of interrelated tasks all aimed at maintaining an acceptable level of confidence that security protections are, in fact, working as intended. These tasks are assisted by data from automated instrumentation of system and application functions.

Data gathered from a secured environment is used to accumulate evidence that security systems are working as intended and to detect incidents and patterns of misuse for further actions. Once messages have been collected, various reports may be created in support of security assurance and administration information requirements.

When a site runs multiple heterogeneous applications, each application system may have its own security mechanisms - user log-on, roles, access right permissions and restrictions, etc. Each application system also has its own security log file that records security relevant events, e.g., log-in, data access, and updates to the security policy databases. A system administrator or security auditor must examine each of these log files to find security

relevant incidents. Not only is it difficult to examine each of these files separately, the format and contents of each file may be confusingly different.

Resolving these issues requires a framework to:

- Maximize interoperability and the meaningfulness of data across applications and sites
- Minimize ambiguity among heterogeneous systems
- Simplify and limit the costs of administrative audit tasks.

3.2. Efficiency

One of the leading concerns about auditing is the potential volume of data gathering and its impact on application system performance. Although this document does not prescribe specific implementations or strategies, the following are meant as informative guidance for development.

1) Audits should be created for transactions or record-level data access, not for individual attribute-level changes to data.

2) This document does not discourage locally optimized gathering of audit data on each application system. Instead, it anticipates implementation-defined periodic gathering and transmission of data to a common repository. This common repository would be optimized for after-the-fact audit queries and reporting, thus unburdening each application system of those responsibilities. It is also important to keep the message size compact so that audit data will not penalize normal network operation.

3) On each application system, a variety of policy-based methods could be employed to optimize data gathering and storage, e.g., selective auditing of only events defined as important plus workload buffering and balancing. Data gathering itself should be stateless to avoid the overhead of transactional semantics. In addition, prior to transmission, some filtering, aggregation, and summarization of repeated events would reduce the number of messages. Audit data storage and integrity on each application system need only be scaled for relatively low-volume and short-duration requirements, yet be consistent with implementation-defined minimums for holding the data for subsequent collection.

4) Leveraging existing data collection should be considered. For example, most commercial security subsystems record events in a local common log file, so the log file data can be extracted for communication to a common repository. Also, it is common in some systems' designs to have a transaction log for data reconstruction

in event of database loss, so collecting data-update audit data within this subsystem could reduce impact on application system performance.

5) A security audit repository would gather all audit message data from the different applications in one database with one standard structure. This would allow easier evaluation and querying. Once a suspicious pattern has been found in the audit log repository, investigation might proceed with more detail in the application specific audit log. The presence of a common repository also simplifies and streamlines the implementation of policies for audit data storage, integrity, retention, and destruction.

4. Trigger Events

The following identifies representative trigger events for generating audit messages. This is not a complete list of trigger events.

For those events arising in the security infrastructure the "minimal" and "basic" level of auditing as outlined in the Common Criteria [ISO15408-2] should be used as a reference standard.

4.1. Security Administration

This group includes all actions that create, maintain, query, and display definitions for securing data, functions, and the associated access policies. For each trigger type, the creation, update or amendment, deletion, and activation or deactivation are auditable.

4.1.1. Data Definition

This includes creation, modification, deletion, query, and display of security attributes for data sets, data groups, or classes plus their atomic data elements or attributes.

4.1.2. Function Definition

This includes, for example, creation, modification, deletion, query, or display of security attributes and auditable events for the application functions used for patient management, clinical processes, registry of business objects and methods, program creation and maintenance, etc.

4.1.3. Domain Definition

This includes all activities to create, modify, delete, query, or display security domains according to various organizational categories such as entity-wide, institutional, departmental, etc.

4.1.4. Classification Definition

This includes all activities that create, modify, delete, query or display security categories or groupings for functions and data such as patient management, nursing, clinical, etc.

4.1.5. Permission Definition

This includes all activities that create, modify, delete, query or display the allowable access permissions associated with functions and data, such as create, read, update, delete, and execution of specific functional units or object access or manipulation methods.

4.1.6. Role Definition

This includes all activities that create, modify, delete, query or display security roles according to various task-grouping categories such as security administration, admissions desk, nurses, physicians, clinical specialists, etc. It also includes the association of permissions with roles for role-based access control.

4.1.7. User Definition

This includes all activities that create, modify, delete, query, or display user accounts. It includes password or other authentication data. It also includes the association of roles with users for role-based access control, or permissions with users for user-based access control.

4.2. Audit Administration and Data Access

This category includes all actions that determine the collection and availability of audit data.

4.2.1. Auditable Event Enable or Disable

This reflects a basic policy decision that an event should or should not be audited. Some, but not necessarily all, triggers or use cases must create an audit record. The selection of what to audit depends on administrative policy decisions. Note that, for integrity, this event should always be audited.

4.2.2. Audit Data Access

This includes instances where audit data is viewed or reported for any purpose. Since the audit data itself may include data protected by institutional privacy policies and expose the implementation of those policies, access to the data is highly sensitive. This event should therefore always be audited.

4.2.3. Audit Data Modify or Delete

This includes instances where audit data is modified or deleted. While such operations are sometimes permitted by systems policies, modification or destruction of audit data may well be the result of unauthorized hostile systems access. Therefore, this type of event should always be audited.

4.3. User Access

This category includes events of access to secured data and functions for which audit data might be collected.

4.3.1. Sign-On

This includes successful and unsuccessful attempts from human users and automated system. It also includes re-authentication actions and re-issuing time-sensitive credentials such as Kerberos tickets.

4.3.2. Sign-Off

This includes explicit sign-off events and session abandonment timeouts from human users and automated systems.

4.3.3. Function Access

This includes user invocation of application or system functions that have permission definitions associated with them. Note that in a Discretionary Access Control environment not all functions require permissions, especially if their impact is benign in relation to security policies.

The following are examples of trigger events relevant to healthcare privacy. The actual triggers for institutional data access, policies for non-care functions, and support regulatory requirements need to be identified by application-domain standards developers and system implementers.

4.3.3.1. Subject of Care Record Access

This includes all functions which manipulate basic patient data:

- Create, e.g., demographics or patient profile
- Assign identifier, e.g., medical record number
- Update, amend
- Merge/unmerge, e.g., combine multiple medical records for one patient
- Import/export of data from/to an external source, including printing and creation of portable media copies.
- Delete, e.g., invalid creation of care record

4.3.3.2. Encounter or Visit

This includes all functions which associate a subject of care with an instance of care:

- Create, e.g., demographics or patient profile
- Assign encounter identifier
- Per-admit
- Admit
- Update, amend
- Delete, e.g., invalid creation of encounter record, breakdown of equipment, patient did not arrive as expected

4.3.3.3. Care Protocols

This includes all functions which associate care plans or similar protocols with an instance or subject of care:

- Schedule, initiate
- Update, amend
- Complete
- Cancel

4.3.3.4. Episodes or Problems

This includes specific clinical episodes within an instance of care. Initiate:

- Update, amend
- Resolve, complete
- Cancel

4.3.3.5. Orders and Order Sets

This includes clinical or supplies orders within an instance or episode of care:

- Initiate
- Update, amend
- Check for contraindications
- Verify
- Deliver/complete - including instructions
- Cancel

4.3.3.6. Health Service Event or Act

This includes various health services scheduled and performed within an instance or episode of care:

- Schedule, initiate
- Update, amend
- Check for contraindications
- Verify
- Perform/complete - including instructions
- Cancel

4.3.3.7. Medications

This includes all medication orders and administration within an instance or episode of care:

- Order
- Check
- Check for interactions
- Verify
- Dispense/deliver - including administration instructions
- Administer
- Cancel

4.3.3.8. Staff/Participant Assignment

This includes staffing or participant assignment actions relevant to an instance or episode of care:

- Assignment of healthcare professionals, caregivers attending physician, residents, medical students, consultants, etc.
- Change in assigned role or authorization, e.g., relative to healthcare status change.
- De-assignment

5. Data Definitions

This section defines and describes the data in the XML schema. The actual XML schema definition is in section 6.

The proposed data elements are grouped into these categories:

1) Event Identification - what was done
2) Active Participant Identification - by whom
3) Network Access Point Identification - initiated from where
4) Audit Source Identification - using which server
5) Participant Object Identification - to what record

5.1. Event Identification

The following data identifies the name, action type, time, and disposition of the audited event. There is only one set of event identification data per audited event.

5.1.1. Event ID

Description

Identifier for a specific audited event, e.g., a menu item, program, rule, policy, function code, application name, or URL. It identifies the performed function.

Optionality: Required

Format / Values

Coded value, either defined by the system implementers or as a reference to a standard vocabulary. The "code" attribute must be unambiguous and unique, at least within Audit Source ID (see section 5.4). Examples of Event IDs are program name, method name, or function name.

For implementation defined coded values or references to standards, the XML schema defines these optional attributes:

Attribute	Value
CodeSystem	OID reference
CodeSystemName	Name of the coding system; strongly recommended to be valued for locally-defined code-sets.
DisplayName	The value to be used in displays and reports
OriginalText	Input value that was translated to the code

To support the requirement for unambiguous event identification, multiple values may not be specified.

Rationale

This identifies the audited function. For "Execute" Event Action Code audit records, this identifies the application function performed.

5.1.2. Event Action Code

Description

Indicator for type of action performed during the event that generated the audit.

Optionality: Optional

Format / Values

Enumeration:

Value	Meaning	Examples
C	Create	Create a new database object, such as Placing an Order.
R	Read/View/Print/Query	Display or print data, such as a Doctor Census
U	Update	Update data, such as Revise Patient Information
D	Delete	Delete items, such as a doctor master file record
E	Execute	Perform a system or application function such as log-on, program execution, or use of an object's method

Rationale

This broadly indicates what kind of action was done on the Participant Object.

Notes

Actions that are not enumerated above are considered an Execute of a specific function or object interface method or treated two or more distinct events. An application action, such as an authorization, is a function Execute, and the Event ID would identify the function.

For some applications, such as radiological imaging, a Query action may only determine the presence of data but not access the data itself. Auditing need not make as fine a distinction.

Compound actions, such as "Move," would be audited by creating audit data for each operation - read, create, delete - or as an Execute of a function or method.

5.1.3. Event Date/Time

Description

Universal coordinated time (UTC), i.e., a date/time specification that is unambiguous as to local time zones.

Optionality: Required

Format / Values

A date/time representation that is unambiguous in conveying universal coordinated time (UTC), formatted according to the ISO 8601 standard [ISO8601]

Rationale

This ties an event to a specific date and time. Security audits typically require a consistent time base, e.g., UTC, to eliminate time-zone issues arising from geographical distribution.

Notes

In a distributed system, some sort of common time base, e.g., an NTP [RFC1305] server, is a good implementation tactic.

5.1.4. Event Outcome Indicator

Description

Indicates whether the event succeeded or failed.

Optionality: Required

Format / Values

Enumeration:

Value	Meaning
0	Success
4	Minor failure; action restarted, e.g., invalid password with first retry
8	Serious failure; action terminated, e.g., invalid password with excess retries
12	Major failure; action made unavailable, e.g., user account disabled due to excessive invalid log-on attempts

Rationale

Some audit events may be qualified by success or failure indicator. For example, a Log-on might have this flag set to a non-zero value to indicate why a log-on attempt failed.

Notes

In some cases a "success" may be partial, for example, an incomplete or interrupted transfer of a radiological study. For the purpose of establishing accountability, these distinctions are not relevant.

5.1.5. Event Type Code

Description

Identifier for the category of event.

Optionality: Optional

Format / Values

Coded value enumeration, either defined by the system implementers or as a reference to a standard vocabulary. For implementation defined codes or references to standards, the XML schema defines these optional attributes:

Attribute	Value
CodeSystem	OID reference
CodeSystemName	Name of the coding system; strongly recommended to be valued for locally-defined code-sets.
DisplayName	The value to be used in displays and reports
OriginalText	Input value that was translated to the code

Since events may be categorized in more than one way, there may be multiple values specified.

Rationale

This field enables queries of messages by implementation-defined event categories.

5.2. Active Participant Identification

The following data identify a user for the purpose of documenting accountability for the audited event. A user may be a person, or a hardware device or software process for events that are not initiated by a person.

Optionally, the user's network access location may be specified.

There may be more than one user per event, for example, in cases of actions initiated by one user for other users, or in events that involve more than one user, hardware device, or system process. However, only one user may be the initiator/requestor for the event.

5.2.1. User ID

Description

Unique identifier for the user actively participating in the event

Optionality: Required

Format / Values

User identifier text string from the authentication system. It is a unique value within the Audit Source ID (see section 5.4).

Rationale

This field ties an audit event to a specific user.

Notes

For cross-system audits, especially with long retention, this user identifier will permanently tie an audit event to a specific user via a perpetually unique key.

For node-based authentication -- where only the system hardware or process, but not a human user, is identified -- User ID would be the node name.

5.2.2. Alternative User ID

Description

Alternative unique identifier for the user

Optionality: Optional

Format / Values

User identifier text string from authentication system. This identifier would be one known to a common authentication system (e.g., single sign-on), if available.

Rationale

In some situations a user may authenticate with one identity but, to access a specific application system, may use a synonymous identify. For example, some "single sign on" implementations will do this. The alternative identifier would then be the original identify used for authentication, and the User ID is the one known to and used by the application.

5.2.3. User Name

Description

The human-meaningful name for the user

Optionality: Optional

Format / Values

Text string

Rationale

The User ID and Alternative User ID may be internal or otherwise obscure values. This field assists the auditor in identifying the actual user.

5.2.4. User Is Requestor

Description

Indicator that the user is or is not the requestor, or initiator, for the event being audited.

Optionality: Optional

Format / Values

Boolean, default/assumed value is "true"

Rationale

This value is used to distinguish between requestor-users and recipient-users. For example, one person may initiate a report-output to be sent to a another user.

5.2.5. Role ID Code

Description

Specification of the role(s) the user plays when performing the event, as assigned in role-based access control security.

Optionality: Optional; multi-valued

Format / Values

Coded value, with attribute "code" valued with the role code or text from authorization system. More than one value may be specified.

The codes may be implementation-defined or reference a standard vocabulary enumeration. For implementation defined codes or references to standards, the XML schema defines these optional attributes:

Attribute	Value description
CodeSystem	OID reference
CodeSystemName	Name of the coding system; strongly recommended to be valued for locally-defined code-sets.
Display Name	The value to be used in displays and reports
OriginalText	Input value that was translated to the code

Rationale

This value ties an audited event to a user's role(s). It is an optional value that might be used to group events for analysis by user functional role categories.

Notes

Many security systems are unable to produce this data, hence it is optional.

For the common message, this identifier would be the one known to a common authorization system, if available. Otherwise, it is a unique value within the Audit Source ID (see section 5.4). Consider using a globally unique identifier associated with the role to avoid ambiguity in auditing data collected from multiple systems.

Role ID is not a substitute for personal accountability.

Ambiguities arise from composite roles and users with multiple roles, i.e., which role within a composite is being used or what privilege was a user employing?

5.3. Network Access Point Identification

The network access point identifies the logical network location for application activity. These data are paired 1:1 with the Active Participant Identification data.

5.3.1. Network Access Point Type Code

Description

An identifier for the type of network access point that originated the audit event.

Optionality: Optional

Format / Values

Enumeration:

Value	Meaning
1	Machine Name, including DNS name
2	IP Address
3	Telephone Number

Rationale

This datum identifies the type of network access point identifier of the user device for the audit event. It is an optional value that may be used to group events recorded on separate servers for analysis of access according to a network access point's type.

5.3.2. Network Access Point ID

Description

An identifier for the network access point of the user device for the audit event. This could be a device id, IP address, or some other identifier associated with a device.

Optionality: Optional

Format / Values

Text may be constrained to only valid values for the given Network Access Point Type, if specified. Recommendation is to be as specific as possible where multiple options are available.

Rationale

This datum identifies the user's network access point, which may be distinct from the server that performed the action. It is an optional value that may be used to group events recorded on separate servers for analysis of a specific network access point's data access across all servers.

Note

Network Access Point ID is not a substitute for personal accountability. Internet IP addresses, in particular, are highly volatile and may be assigned to more than one person in a short time period.

Examples

Network Access Point ID: SMH4WC02
Network Access Point Type: 1 = Machine Name

Network Access Point ID: 192.0.2.2
Network Access Point Type: 2 = IP address

Network Access Point ID: 610-555-1212
Network Access Point Type: 3 = Phone Number

5.4. Audit Source Identification

The following data are required primarily for application systems and processes. Since multi-tier, distributed, or composite applications make source identification ambiguous, this collection of fields may repeat for each application or process actively involved in the event. For example, multiple value-sets can identify participating web servers, application processes, and database server threads in an n-tier distributed application. Passive event participants, e.g., low-level network transports, need not be identified.

Depending on implementation strategies, it is possible that the components in a multi-tier, distributed, or composite applications may generate more than one audit message for a single application event. Various data in the audit message may be used to identify such cases, supporting subsequent data reduction. This document anticipates that the repository and reporting mechanisms will perform data reduction when required, but does not specify those mechanism.

5.4.1. Audit Enterprise Site ID

Description

Logical source location within the healthcare enterprise network, e.g., a hospital or other provider location within a multi-entity provider group.

Optionality: Optional

Format / Values

Unique identifier text string within the healthcare enterprise. May be unvalued when the audit-generating application is uniquely identified by Audit Source ID.

Rationale

This value differentiates among the sites in a multi-site enterprise health information system.

Notes

This is defined by the application that generates the audit record. It contains a unique code that identifies a business organization (owner of data) that is known to the enterprise. The value further qualifies and disambiguates the Audit Source ID. Values may vary depending on type of business. There may be levels of differentiation within the organization.

5.4.2. Audit Source ID

Description

Identifier of the source where the event originated.

Optionality: Required

Format / Values

Unique identifier text string, at least within the Audit Enterprise Site ID

Rationale

This field ties the event to a specific source system. It may be used to group events for analysis according to where the event occurred.

Notes

In some configurations, a load-balancing function distributes work among two or more duplicate servers. The values defined for this field thus may be considered as an source identifier for a group of servers rather than a specific source system.

5.4.3. Audit Source Type Code

Description

Code specifying the type of source where event originated.

Optionality: Optional

Format / Values

Coded-value enumeration, optionally defined by system implementers or a as a reference to a standard vocabulary. Unless defined or referenced, the default values for the "code" attribute are:

Value	Meaning
1	End-user interface
2	Data acquisition device or instrument
3	Web server process tier in a multi-tier system
4	Application server process tier in a multi-tier system
5	Database server process tier in a multi-tier system
6	Security server, e.g., a domain controller
7	ISO level 1-3 network component
8	ISO level 4-6 operating software
9	External source, other or unknown type

For implementation defined codes or references to standards, the XML schema defines these optional attributes:

Attribute	Value
CodeSystem	OID reference
CodeSystemName	Name of the coding system; strongly recommended to be valued for locally-defined code-sets.
DisplayName	The value to be used in displays and reports
OriginalText	Input value that was translated to the code

Since audit sources may be categorized in more than one way, there may be multiple values specified.

Rationale

This field indicates which type of source is identified by the Audit Source ID. It is an optional value that may be used to group events for analysis according to the type of source where the event occurred.

5.5. Participant Object Identification

The following data assist the auditing process by indicating specific instances of data or objects that have been accessed.

These data are required unless the values for Event Identification, Active Participant Identification, and Audit Source Identification are sufficient to document the entire auditable event. Production of

audit records containing these data may be enabled or suppressed, as determined by healthcare organization policy and regulatory requirements.

Because events may have more than one participant object, this group can be a repeating set of values. For example, depending on institutional policies and implementation choices:

- Two participant object value-sets can be used to identify access to patient data by medical record number plus the specific health care encounter or episode for the patient.
- A patient participant and his authorized representative may be identified concurrently.
- An attending physician and consulting referrals may be identified concurrently.
- All patients identified on a worklist may be identified.
- For radiological studies, a set of related participant objects identified by accession number or study number, may be identified.

Note, though, that each audit message documents only a single usage instance of such participant object relationships and does not serve to document all relationships that may be present or possible.

5.5.1. Participant Object Type Code

Description

Code for the participant object type being audited. This value is distinct from the user's role or any user relationship to the participant object.

Optionality: Optional

Format / Values

Enumeration:

Value	Meaning
1	Person
2	System Object
3	Organization
4	Other

Rationale

To describe the object being acted upon. In addition to queries on the subject of the action in an auditable event, it is also important to be able to query on the object type for the action.

5.5.2. Participant Object Type Code Role

Description

Code representing the functional application role of Participant Object being audited

Optionality: Optional

Format / Values

Enumeration, specific to Participant Object Type Code:

Value	Meaning	Participant Object Type Codes
1	Patient	1 - Person
2	Location	3 - Organization
3	Report	2 - System Object
4	Resource	1 - Person 3 - Organization
5	Master file	2 - System Object
6	User	1 - Person 2 - System Object (non-human user)
7	List	2 - System Object
8	Doctor	1 - Person
9	Subscriber	3 - Organization
10	Guarantor	1 - Person 3 - Organization
11	Security User Entity	1 - Person 2 - System Object
12	Security User Group	2 - System Object
13	Security Resource	2 - System Object
14	Security Granularity Definition	2 - System Object
15	Provider	1 - Person 3 - Organization
16	Data Destination	2 - System Object
17	Data Repository	2 - System Object
18	Schedule	2 - System Object
19	Customer	3 - Organization
20	Job	2 - System Object
21	Job Stream	2 - System Object

22	Table	2 - System Object
23	Routing Criteria	2 - System Object
24	Query	2 - System Object

A "Security Resource" is an abstract securable object, e.g., a screen, interface, document, program, etc. -- or even an audit data set or repository.

Rationale

For some detailed audit analysis it may be necessary to indicate a more granular type of participant, based on the application role it serves.

5.5.3. Participant Object Data Life Cycle

Description

Identifier for the data life-cycle stage for the participant object. This can be used to provide an audit trail for data, over time, as it passes through the system.

Optionality: Optional

Format/Values

Enumeration:

Value	Meaning
1	Origination / Creation
2	Import / Copy from original
3	Amendment
4	Verification
5	Translation
6	Access / Use
7	De-identification
8	Aggregation, summarization, derivation
9	Report
10	Export / Copy to target
11	Disclosure
12	Receipt of disclosure
13	Archiving
14	Logical deletion
15	Permanent erasure / Physical destruction

Rationale

Institutional policies for privacy and security may optionally fall under different accountability rules based on data life cycle. This provides a differentiating value for those cases.

5.5.4. Participant Object ID Type Code

Description

Describes the identifier that is contained in Participant Object ID.

Optionality: Required

Format / Values

Coded-value enumeration, specific to Participant Object Type Code, using attribute-name "code". The codes below are the default set.

Value	Meaning	Participant Object Type Codes
1	Medical Record Number	1 - Person
2	Patient Number	1 - Person
3	Encounter Number	1 - Person
4	Enrollee Number	1 - Person
5	Social Security Number	1 - Person
6	Account Number	1 - Person
		3 - Organization
7	Guarantor Number	1 - Person
		3 - Organization
8	Report Name	2 - System Object
9	Report Number	2 - System Object
10	Search Criteria	2 - System Object
11	User Identifier	1 - Person
		2 - System Object
12	URI	2 - System Object

User Identifier and URI [RFC2396] text strings are intended to be used for security administration trigger events to identify the objects being acted-upon.

The codes may be the default set stated above, implementation-defined, or reference a standard vocabulary enumeration, such as HL7 version 2.4 table 207 or DICOM defined media types. For implementation defined codes or references to standards, the XML schema defines these optional attributes:

Attribute	Value
CodeSystem	OID reference
CodeSystemName	Name of the coding system; strongly recommended to be valued for locally-defined code-sets.
DisplayName	The value to be used in displays and reports
OriginalText	Input value that was translated to the code

Rationale

Required to distinguish among various identifiers that may synonymously identify a participant object.

5.5.5. Participant Object Sensitivity

Description

Denotes policy-defined sensitivity for the Participant Object ID such as VIP, HIV status, mental health status, or similar topics.

Optionality: Optional

Format / Values

Values are institution- and implementation-defined text strings.

5.5.6. Participant Object ID

Description

Identifies a specific instance of the participant object.

Optionality: Required

Format / Values

Text string. Value format depends on Participant Object Type Code and the Participant Object ID Type Code.

Rationale

This field identifies a specific instance of an object, such as a patient, to detect/track privacy and security issues.

Notes

Consider this to be the primary unique identifier key for the object, so it may be a composite data field as implemented.

5.5.7. Participant Object Name

Description

An instance-specific descriptor of the Participant Object ID audited, such as a person's name.

Optionality: Optional

Format / Values

Text string

Rationale

This field may be used in a query/report to identify audit events for a specific person, e.g., where multiple synonymous Participant Object IDs (patient number, medical record number, encounter number, etc.) have been used.

5.5.8. Participant Object Query

Description

The actual query for a query-type participant object.

Optionality: Optional

Format / Values

Base 64 encoded data

Rationale

For query events it may be necessary to capture the actual query input to the query process in order to identify the specific event. Because of differences among query implementations and data encoding for them, this is a base 64 encoded data blob. It may be subsequently decoded or interpreted by downstream audit analysis processing.

5.5.9. Participant Object Detail

Description

Implementation-defined data about specific details of the object accessed or used.

Optionality: Optional

Format

Type-value pair. The "type" attribute is an implementation-defined text string. The "value" attribute is a base 64 encoded data.

Rationale

Specific details or values from the object accessed may be desired in specific auditing implementations. The type-value pair enables the use of implementation-defined and locally-extensible object type identifiers and values. For example, a clinical diagnostic object may contain multiple test results, and this element could document the type and number and type of results.

Many possible data encodings are possible for this elements, so the value is a base 64 encoded data blob. It may be subsequently decoded or interpreted by downstream audit analysis processing.

6. XML Schema

This section contains the actual XML schema definition for the data defined in section 5. It also provides brief guidance for specifying schema localizations for implementation purposes.

The XML schema specified in section 6.1 conforms with the W3C Recommendations for XML Schema structure [W3CXML-1] and data types [W3CXML-2].

6.1. XML Schema Definition

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
 elementFormDefault="qualified" attributeFormDefault="unqualified">
 <xs:element name="AuditMessage">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="EventIdentification"
     type="EventIdentificationType"/>
    <xs:element name="ActiveParticipant" maxOccurs="unbounded">
     <xs:complexType>
      <xs:complexContent>
       <xs:extension base="ActiveParticipantType"/>
      </xs:complexContent>
     </xs:complexType>
    </xs:element>
    <xs:element name="AuditSourceIdentification"
```

```
     type="AuditSourceIdentificationType" maxOccurs="unbounded"/>
    <xs:element name="ParticipantObjectIdentification"
     type="ParticipantObjectIdentificationType" minOccurs="0"
     maxOccurs="unbounded"/>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
 <xs:complexType name="EventIdentificationType">
  <xs:sequence>
   <xs:element name="EventID" type="CodedValueType"/>
   <xs:element name="EventTypeCode" type="CodedValueType"
    minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="EventActionCode" use="optional">
   <xs:simpleType>
    <xs:restriction base="xs:string">
     <xs:enumeration value="C">
      <xs:annotation>
       <xs:appinfo>Create</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="R">
      <xs:annotation>
       <xs:appinfo>Read</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="U">
      <xs:annotation>
       <xs:appinfo>Update</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="D">
      <xs:annotation>
       <xs:appinfo>Delete</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="E">
      <xs:annotation>
       <xs:documentation>Execute</xs:documentation>
      </xs:annotation>
     </xs:enumeration>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="EventDateTime" type="xs:dateTime"
   use="required"/>
  <xs:attribute name="EventOutcomeIndicator" use="required">
   <xs:simpleType>
```

```
    <xs:restriction base="xs:integer">
     <xs:enumeration value="0">
      <xs:annotation>
       <xs:appinfo>Success</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="4">
      <xs:annotation>
       <xs:appinfo>Minor failure</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="8">
      <xs:annotation>
       <xs:appinfo>Serious failure</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="12">
      <xs:annotation>
       <xs:appinfo>Major failure; action made unavailable
          </xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
 </xs:complexType>
 <xs:complexType name="AuditSourceIdentificationType">
  <xs:sequence>
   <xs:element name="AuditSourceTypeCode" minOccurs="0"
    maxOccurs="unbounded">
    <xs:complexType>
     <xs:complexContent>
      <xs:restriction base="CodedValueType">
       <xs:attribute name="code" use="required">
        <xs:simpleType>
         <xs:restriction base="xs:string">
          <xs:enumeration value="1">
           <xs:annotation>
            <xs:appinfo>End-user display device, diagnostic
             display</xs:appinfo>
           </xs:annotation>
          </xs:enumeration>
          <xs:enumeration value="2">
           <xs:annotation>
            <xs:appinfo>Data acquisition device or
             instrument</xs:appinfo>
           </xs:annotation>
          </xs:enumeration>
```

```
          <xs:enumeration value="3">
           <xs:annotation>
            <xs:appinfo>Web server process</xs:appinfo>
           </xs:annotation>
          </xs:enumeration>
          <xs:enumeration value="4">
           <xs:annotation>
            <xs:appinfo>Application server process</xs:appinfo>
           </xs:annotation>
          </xs:enumeration>
          <xs:enumeration value="5">
           <xs:annotation>
            <xs:appinfo>Database server process</xs:appinfo>
           </xs:annotation>
          </xs:enumeration>
          <xs:enumeration value="6">
           <xs:annotation>
            <xs:appinfo>Security server, e.g., a domain
             controller</xs:appinfo>
           </xs:annotation>
          </xs:enumeration>
          <xs:enumeration value="7">
           <xs:annotation>
            <xs:documentation>ISO level 1-3 network
             component</xs:documentation>
           </xs:annotation>
          </xs:enumeration>
          <xs:enumeration value="8">
           <xs:annotation>
            <xs:appinfo>ISO level 4-6 operating software</xs:appinfo>
           </xs:annotation>
          </xs:enumeration>
          <xs:enumeration value="9">
           <xs:annotation>
            <xs:appinfo>External source, other or unknown
             type</xs:appinfo>
           </xs:annotation>
          </xs:enumeration>
         </xs:restriction>
        </xs:simpleType>
       </xs:attribute>
      </xs:restriction>
     </xs:complexContent>
    </xs:complexType>
   </xs:element>
  </xs:sequence>
  <xs:attribute name="AuditEnterpriseSiteID" type="xs:string"
   use="optional"/>
```

```
  <xs:attribute name="AuditSourceID" type="xs:string"
   use="required"/>
 </xs:complexType>
 <xs:complexType name="ActiveParticipantType">
  <xs:sequence minOccurs="0">
   <xs:element name="RoleIDCode" type="CodedValueType" minOccurs="0"
    maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="UserID" type="xs:string" use="required"/>
  <xs:attribute name="AlternativeUserID" type="xs:string"
   use="optional"/>
  <xs:attribute name="UserName" type="xs:string" use="optional"/>
  <xs:attribute name="UserIsRequestor" type="xs:boolean"
   use="optional" default="true"/>
  <xs:attribute name="NetworkAccessPointID" type="xs:string"
   use="optional"/>
  <xs:attribute name="NetworkAccessPointTypeCode" use="optional">
   <xs:simpleType>
    <xs:restriction base="xs:unsignedByte">
     <xs:enumeration value="1">
      <xs:annotation>
       <xs:appinfo>Machine Name, including DNS name</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="2">
      <xs:annotation>
       <xs:appinfo>IP Address</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="3">
      <xs:annotation>
       <xs:appinfo>Telephone Number</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
 </xs:complexType>
 <xs:complexType name="ParticipantObjectIdentificationType">
  <xs:sequence>
   <xs:element name="ParticipantObjectIDTypeCode">
    <xs:complexType>
     <xs:complexContent>
      <xs:restriction base="CodedValueType">
       <xs:attribute name="code" use="required">
        <xs:simpleType>
         <xs:restriction base="xs:string">
          <xs:enumeration value="1">
```

```
            <xs:annotation>
             <xs:appinfo>Medical Record Number</xs:appinfo>
            </xs:annotation>
           </xs:enumeration>
           <xs:enumeration value="2">
            <xs:annotation>
             <xs:appinfo>Patient Number</xs:appinfo>
            </xs:annotation>
           </xs:enumeration>
           <xs:enumeration value="3">
            <xs:annotation>
             <xs:appinfo>Encounter Number</xs:appinfo>
            </xs:annotation>
           </xs:enumeration>
           <xs:enumeration value="4">
            <xs:annotation>
             <xs:appinfo>Enrollee Number</xs:appinfo>
            </xs:annotation>
           </xs:enumeration>
           <xs:enumeration value="5">
            <xs:annotation>
             <xs:appinfo>Social Security Number</xs:appinfo>
            </xs:annotation>
           </xs:enumeration>
           <xs:enumeration value="6">
            <xs:annotation>
             <xs:appinfo>Account Number</xs:appinfo>
            </xs:annotation>
           </xs:enumeration>
           <xs:enumeration value="7">
            <xs:annotation>
             <xs:appinfo>Guarantor Number</xs:appinfo>
            </xs:annotation>
           </xs:enumeration>
           <xs:enumeration value="8">
            <xs:annotation>
             <xs:appinfo>Report Name</xs:appinfo>
            </xs:annotation>
           </xs:enumeration>
           <xs:enumeration value="9">
            <xs:annotation>
             <xs:appinfo>Report Number</xs:appinfo>
            </xs:annotation>
           </xs:enumeration>
           <xs:enumeration value="10">
            <xs:annotation>
             <xs:appinfo>Search Criteria</xs:appinfo>
            </xs:annotation>
```

```
          </xs:enumeration>
          <xs:enumeration value="11">
           <xs:annotation>
            <xs:appinfo>User Identifier</xs:appinfo>
           </xs:annotation>
          </xs:enumeration>
          <xs:enumeration value="12">
           <xs:annotation>
            <xs:appinfo>URI</xs:appinfo>
           </xs:annotation>
          </xs:enumeration>
          <xs:enumeration value=""/>
         </xs:restriction>
        </xs:simpleType>
       </xs:attribute>
      </xs:restriction>
     </xs:complexContent>
    </xs:complexType>
   </xs:element>
   <xs:choice minOccurs="0">
    <xs:element name="ParticipantObjectName" type="xs:string"
     minOccurs="0"/>
    <xs:element name="ParticipantObjectQuery" type="xs:base64Binary"
     minOccurs="0"/>
   </xs:choice>
   <xs:element name="ParticipantObjectDetail"
    type="TypeValuePairType" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="ParticipantObjectID" type="xs:string"
   use="required"/>
  <xs:attribute name="ParticipantObjectTypeCode" use="optional">
   <xs:simpleType>
    <xs:restriction base="xs:unsignedByte">
     <xs:enumeration value="1">
      <xs:annotation>
       <xs:appinfo>Person</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="2">
      <xs:annotation>
       <xs:appinfo>System object</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="3">
      <xs:annotation>
       <xs:appinfo>Organization</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
```

```
     <xs:enumeration value="4">
      <xs:annotation>
       <xs:appinfo>Other</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="ParticipantObjectTypeCodeRole" use="optional">
   <xs:simpleType>
    <xs:restriction base="xs:unsignedByte">
     <xs:enumeration value="1">
      <xs:annotation>
       <xs:appinfo>Patient</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="2">
      <xs:annotation>
       <xs:appinfo>Location</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="3">
      <xs:annotation>
       <xs:appinfo> Report</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="4">
      <xs:annotation>
       <xs:appinfo>Resource</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="5">
      <xs:annotation>
       <xs:appinfo>Master file</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="6">
      <xs:annotation>
       <xs:appinfo>User</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="7">
      <xs:annotation>
       <xs:appinfo>List</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="8">
      <xs:annotation>
```

```
       <xs:appinfo>Doctor</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="9">
      <xs:annotation>
       <xs:appinfo>Subscriber</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="10">
      <xs:annotation>
       <xs:appinfo>Guarantor</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="11">
      <xs:annotation>
       <xs:appinfo>Security User Entity</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="12">
      <xs:annotation>
       <xs:appinfo>Security User Group</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="13">
      <xs:annotation>
       <xs:appinfo>Security Resource</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="14">
      <xs:annotation>
       <xs:appinfo>Security Granualarity Definition</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="15">
      <xs:annotation>
       <xs:appinfo>Provider</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="16">
      <xs:annotation>
       <xs:appinfo>Report Destination</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="17">
      <xs:annotation>
       <xs:appinfo>Report Library</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
```

```
     <xs:enumeration value="18">
      <xs:annotation>
       <xs:appinfo>Schedule</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="19">
      <xs:annotation>
       <xs:appinfo>Customer</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="20">
      <xs:annotation>
       <xs:appinfo>Job</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="21">
      <xs:annotation>
       <xs:appinfo>Job Stream</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="22">
      <xs:annotation>
       <xs:appinfo>Table</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="23">
      <xs:annotation>
       <xs:appinfo>Routing Criteria</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="24">
      <xs:annotation>
       <xs:appinfo>Query</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="ParticipantObjectDataLifeCycle" use="optional">
   <xs:simpleType>
    <xs:restriction base="xs:unsignedByte">
     <xs:enumeration value="1">
      <xs:annotation>
       <xs:appinfo>Origination / Creation</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="2">
      <xs:annotation>
```

```
       <xs:appinfo>Import / Copy from original </xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="3">
      <xs:annotation>
       <xs:appinfo>Amendment</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="4">
      <xs:annotation>
       <xs:appinfo>Verification</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="5">
      <xs:annotation>
       <xs:appinfo>Translation</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="6">
      <xs:annotation>
       <xs:appinfo>Access / Use</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="7">
      <xs:annotation>
       <xs:appinfo>De-identification</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="8">
      <xs:annotation>
       <xs:appinfo>Aggregation, summarization,
        derivation</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="9">
      <xs:annotation>
       <xs:appinfo>Report</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="10">
      <xs:annotation>
       <xs:appinfo>Export / Copy to target</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="11">
      <xs:annotation>
       <xs:appinfo>Disclosure</xs:appinfo>
      </xs:annotation>
```

```
     </xs:enumeration>
     <xs:enumeration value="12">
      <xs:annotation>
       <xs:appinfo>Receipt of disclosure</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="13">
      <xs:annotation>
       <xs:appinfo>Archiving</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="14">
      <xs:annotation>
       <xs:appinfo>Logical deletion</xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
     <xs:enumeration value="15">
      <xs:annotation>
       <xs:appinfo>Permanent erasure / Physical destruction
       </xs:appinfo>
      </xs:annotation>
     </xs:enumeration>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="ParticipantObjectSensitivity" type="xs:string"
   use="optional"/>
 </xs:complexType>
 <xs:complexType name="CodedValueType">
  <xs:attribute name="code" type="xs:string" use="required"/>
  <xs:attributeGroup ref="CodeSystem"/>
  <xs:attribute name="displayName" type="xs:string" use="optional"/>
  <xs:attribute name="originalText" type="xs:string" use="optional"/>
 </xs:complexType>
 <xs:complexType name="TypeValuePairType">
  <xs:attribute name="type" type="xs:string" use="required"/>
  <xs:attribute name="value" type="xs:base64Binary" use="required"/>
 </xs:complexType>
 <xs:attributeGroup name="CodeSystem">
  <xs:attribute name="codeSystem" type="OID" use="optional"/>
  <xs:attribute name="codeSystemName" type="xs:string"
   use="optional"/>
 </xs:attributeGroup>
 <xs:simpleType name="OID">
  <xs:restriction base="xs:string">
   <xs:whiteSpace value="collapse"/>
  </xs:restriction>
 </xs:simpleType>
```

```
</xs:schema>
```

6.2. XML Schema Localization

The schema specified in section 6.1 may be extended and restricted to meet local implementation-specific requirements. W3C Recommendation for XML Schema structure [W3CXML-1], section 4, is the governing standard for accomplishing this.

As of the current version of this document, a public reference URI for the base schema has not been established.

Local definitions reference the common audit message base schema. For example, here is a schema with a local vocabulary restriction for "Audit Enterprise Site ID" plus an extension adding a new "Audit Source Asset Number" element.

The URI used to identify this schema (http://audit-message-uri) is a syntactically valid example that does not represent an actual schema. Schema validators might report an error when attempting to import a schema using this URI.

```
<xs:schema xmlns:audit="http://audit-message-URI"
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 elementFormDefault="qualified" attributeFormDefault="unqualified">
 <xs:import schemaLocation="http://audit-message-URI"/>
 <xs:complexType name="LocaAuditSourceIdentificationType">
  <xs:complexContent>
   <xs:restriction base="AuditSourceIdentificationType">
    <xs:attribute name="AuditEnterpriseSiteID" use="required">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:enumeration value="Main"/>
       <xs:enumeration value="Clinic1"/>
       <xs:enumeration value="Clinic2"/>
       <xs:enumeration value="Radiology"/>
       <xs:enumeration value="Lab"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
   </xs:restriction>
  </xs:complexContent>
 </xs:complexType>
 <xs:element name="LocalAuditSourceIdentification">
  <xs:complexType>
   <xs:complexContent>
    <xs:extension base="LocaAuditSourceIdentificationType">
      <xs:attribute name="AuditSourceAssetNumber" type="xs:string"
```

```
      use="required"/>
    </xs:extension>
   </xs:complexContent>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

7. Security Considerations

Audit data must be secured at least to the same extent as the underlying data and activities being audited. This includes access controls as well as data integrity and recovery functions. This document acknowledges the need for, but does not specify, the policies and technical methods to accomplish this.

It is conceivable that audit data might have unintended uses, e.g., tracking the frequency and nature of system use for productivity measures. ASTM standard E2147-01 [E2147] states, in paragraph 5.3.10, "Prohibit use for other reasons than to enforce security and to detect security breaches in record health information systems, for example, the audits are not to be used to explore activity profiles or movement profiles of employees."

Some audit data arises from security-relevant processes other than data access. These are the trigger events listed in section 4.1 and 4.2 of this document. Audit data, defined in this document, can record the accountabilities for the results of these processes, as part of a complete security implementation. A discussion of the associated authorities, reference standards, and implementation technology choices for the processes is outside the scope of this document.

8. References

8.1. Normative References

[E2147] "E2147-01 Standard Specification for Audit and Disclosure Logs for Use in Health Information Systems", ASTM International, June 2002.

[ISO15408-2] "ISO/IEC 15408:1999 Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements", ISO, August 1999.

[ISO8601] "ISO 8601:2000 Data elements and interchange formats -- Information interchange -- Representation of dates and times", ISO, December 2000.

[RFC1305] Mills, D., "Network Time Protocol (Version 3) Specification, Implementation", RFC 1305, March 1992.

[RFC2396] Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998.

[W3CXML-1] W3C Recommendation "XML Schema Part 1: Structures", version 1.0, May 2001.

[W3CXML-2] W3C Recommendation "XML Schema Part 2: Datatypes," version 1.0, May 2001.

8.2. Informative References

[HL7SASIG] Marshall, G. and G. Dickinson, "Common Audit Message", HL7 Security and Accountability Special Interest Group, November 2001.

[IHETF-3] "IHE Technical Framework", Volume III, HIMMS/RSNA, April 2002.

[NEMASPC] "Security and Privacy Auditing in Health Care Information Technology", Joint NEMA/COCIR/JIRA Security and Privacy Committee, 26 June 2001.

Acknowledgments

The author gratefully acknowledges the advice and assistance of the following people during the preparation of this document:

Carmela Couderc, Siemens Medical Solutions
Michael Davis, SAIC
Gary Dickinson
Christoph Dickmann, Siemens Medical Solutions
Daniel Hannum, Siemens Medical Solutions
Robert Horn, Agfa
James McAvoy, Siemens Medical Solutions
John Moehrke, General Electric Medical Systems
Jennifer Puyenbroek, McKesson Information Solutions
Angela Ray, McKesson Information Solutions
Lawrence Tarbox, Siemens Corporate Research

Author's Address

Glen Marshall
Siemens Medical Solutions Health Services
51 Valley Stream Parkway
Malvern, PA 19312
USA

Phone: (610) 219-3938
EMail: glen.f.marshall@siemens.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.