kinamik

Secure Audit Vault Security Target

Secure Audit Vault Security Target

Table of Contents

1. Introduction 1
   - 1.1. ST and TOE reference 1
   - 1.2. Document terminology 1
   - 1.3. TOE overview 1
     - 1.3.1. Usage and TOE type 1
     - 1.3.2. Major Security features 2
     - 1.3.3. Required software & hardware 2
   - 1.4. TOE description 2
     - 1.4.1. Architecture 2
     - 1.4.2. Security features 4
     - 1.4.3. Security attributes 5
     - 1.4.4. User data 5
2. Conformance claims 6
3. Security objectives 7
4. Extended components definition 8
5. Security requirements 9
   - 5.1. User data flow 9
     - 5.1.1. FDP_ITC.1.kFeed Import of user data without security attributes 9
     - 5.1.2. FDP_ITT.3.kFeed-kSecure Integrity monitoring 9
     - 5.1.3. FDP_ITT.1.kFeed-kSecure Basic internal transfer protection 9
     - 5.1.4. FDP_ACC.1.kFeed Subset access control 10
     - 5.1.5. FDP_ACF.1.kFeed Security attribute based access control 10
     - 5.1.6. FDP_IFC.1.order Subset information flow control 10
     - 5.1.7. FDP_IFF.1.order Simple security attributes 11
     - 5.1.8. FDP_DAU.1 Basic Data Authentication 11
     - 5.1.9. FDP_UCT.1 Basic data exchange confidentiality 11
     - 5.1.10. FDP_IFC.1.encr Subset information flow control 12
     - 5.1.11. FDP_IFF.1.encr Simple security attributes 12
     - 5.1.12. FDP_ETC.2 Export of user data with security attributes 12
     - 5.1.13. FDP_SDI.2 Stored data integrity monitoring and action 13
     - 5.1.14. FDP_ITC.2.kAuditor Import of user data with security attributes 13
     - 5.1.15. FDP_ACC.1.auditor Subset access control 13
     - 5.1.16. FDP_ACF.1.auditor Security attribute based access control 14
     - 5.1.17. FDP_ACC.1.administrator Subset access control 14
     - 5.1.18. FDP_ACF.1.administrator Security attribute based access control 14
   - 5.2. Management 15
     - 5.2.1. FMT_MSA.3 Static attribute initialisation 15
     - 5.2.2. FMT_MSA.1 Management of security attributes 15
     - 5.2.3. FMT_SMR.1 Security roles 15
     - 5.2.4. FIA_UID.2 User identification before any action 16
     - 5.2.5. FMT_SMF.1 Specification of Management Functions 16
     - 5.2.6. FIA_UAU.2 User authentication before any action 16
6. TOE summary specification 17
   - 6.1. Role management, access control and information flow 17
   - 6.2. Import of the user data to TOE 18
   - 6.3. Storage of user data 19
   - 6.4. Integrity verification 19

Chapter 1. Introduction

This document is the security target for Secure Audit Vault, as defined by the Common Criteria process.

1.1. ST and TOE reference

ST Title	Secure Audit Vault Security Target
ST version	$Revision: 20488 $
TOE name	Secure Audit Vault
TOE version	1.3.6

1.2. Document terminology

This section describes terms and acronyms that are used through the documents involved in the Common Criteria evaluation:

TERM	DEFINITION
Hardened server	Server that has been configured to improve its security and to prevent malicious attacks.
Source	Each trusted application or system that generates data intended to be captured by the Kinamik kFeeds.
Record	Data piece generated from a particular source.
User session	The user session is the period of time where a kFeed is connected to a kSecure.
Sequence of records	Sorted list of records sent during a user session established by the kFeed and kSecure.
Audit trail	Logical container for sequences of records. The audit trail has an id.
JCA	Acronym for Java Cryptography Architecture. It is the standard Java™ API for cryptographic operations.
GSS-API	Acronym for Generic Security Services Application Program Interface.
LDAP	Acronym for the Lightweight Directory Access Protocol.
kNotary	This is a synonym for kSecure. kNotary was the name used in previous versions for the kSecure component. Although kNotary is not used anymore it still appears in some functional specification documents.

1.3. TOE overview

1.3.1. Usage and TOE type

The TOE is a software product that provides tamper-evidence properties to an incoming stream of records in real time, at low computational cost and with integrity at the record level. The TOE is intended to be used on a networked environment.

The TOE uses a patent pending cryptographic protocol to sign electronically. This protocol combines standard cryptographic primitives to provide tamper-evidence properties to an incoming stream of records in real time at low computational cost and with integrity at the record level. Unlike off-the-shelf digital signatures, the protocol can pinpoint at which record tampering occurred.

1.3.2. Major Security features

The protocol has several key differentiating features that make it unique when compared with other integrity techniques.

1. **Granularity**. Data integrity is computed record by record. The main advantage of this approach is that in case of tampering the protocol can not only pinpoint the records affected but it can still verify the integrity of the non-tampered records. In most integrity technologies integrity is computed at a file level, typically with files being consolidated daily, and tampering one single line means that the whole file content has to be discarded.

2. **Enhanced security and non repudiation**. The protocol reduces dramatically the time window when data is vulnerable to attacks. With other technologies attackers can tamper data without detection for as long as the file is open, due to integrity being a post-process that is enforced after the file is closed — typically on a daily basis. This time window has a direct impact on the chain custody and hence on non-repudiation of data. By contrast, Kinamik's protocol computes integrity on real time so that time window disappears or at least decreases by orders of magnitude.

3. **Independent integrity auditing**. The protocol has the option of encrypting records as they arrive. Afterward, integrity can be verified without knowing the private key used to encrypt data. In practice, this means that integrity verification is a function independent of actual access to data.

The system has been coded in Java™ so it can run on multiple hardware and software platforms.

1.3.3. Required software & hardware

The system requires at least one general-purpose computer but recommends three for security reasons. The system does not add further hardware requirements other than those imposed by its software requirements.

The system requires the following software that is not part of the TOE itself.

1. Sun® Java™ Runtime Edition 1.5 plus Java Cryptography Extension Unlimited Strength Jurisdiction Policy Files 5.0.

2. Any of the following

   - Oracle® Database 10g, or

   - Sun® MySQL™ Community Server 5.1.

3. Any of the following

   - Mozilla Firefox® 3, or

   - Microsoft® Internet Explorer® 7.

4. Optionally, an external LDAP server acting as an authentication provider.

5. Optionally, a cryptographic provider other than the default provided with the Java Runtime Edition above.

1.4. TOE description

1.4.1. Architecture

From a logical perspective, there are several sub-systems involved in the data flow.

1. **kSecure**. This component receives data from the kFeeds, signs it using the protocol, and stores both the data and the signature in the database. It is deployed as a web server.

2. **kAuditor**. At user request, this component reads data from the database, verifies its signature and produces an integrity report with the list of tampers (if any) at the record level. It is deployed as a web server and users operate with it using a web GUI.

3. **Database**. A database management system that stores both data and signatures, plus some related meta data. It is important to note that at this point data has been rendered tamper evident by kSecure.

4. **kFeeds**. Independent programs and libraries that collect data. Depending on the data source, they are deployed as standalone programs (agentless kFeed) or embedded as libraries in a third-party system (collocated kFeed).

For the purpose of this document, the TOE comprises only kSecure, kAuditor and the kFeeds.

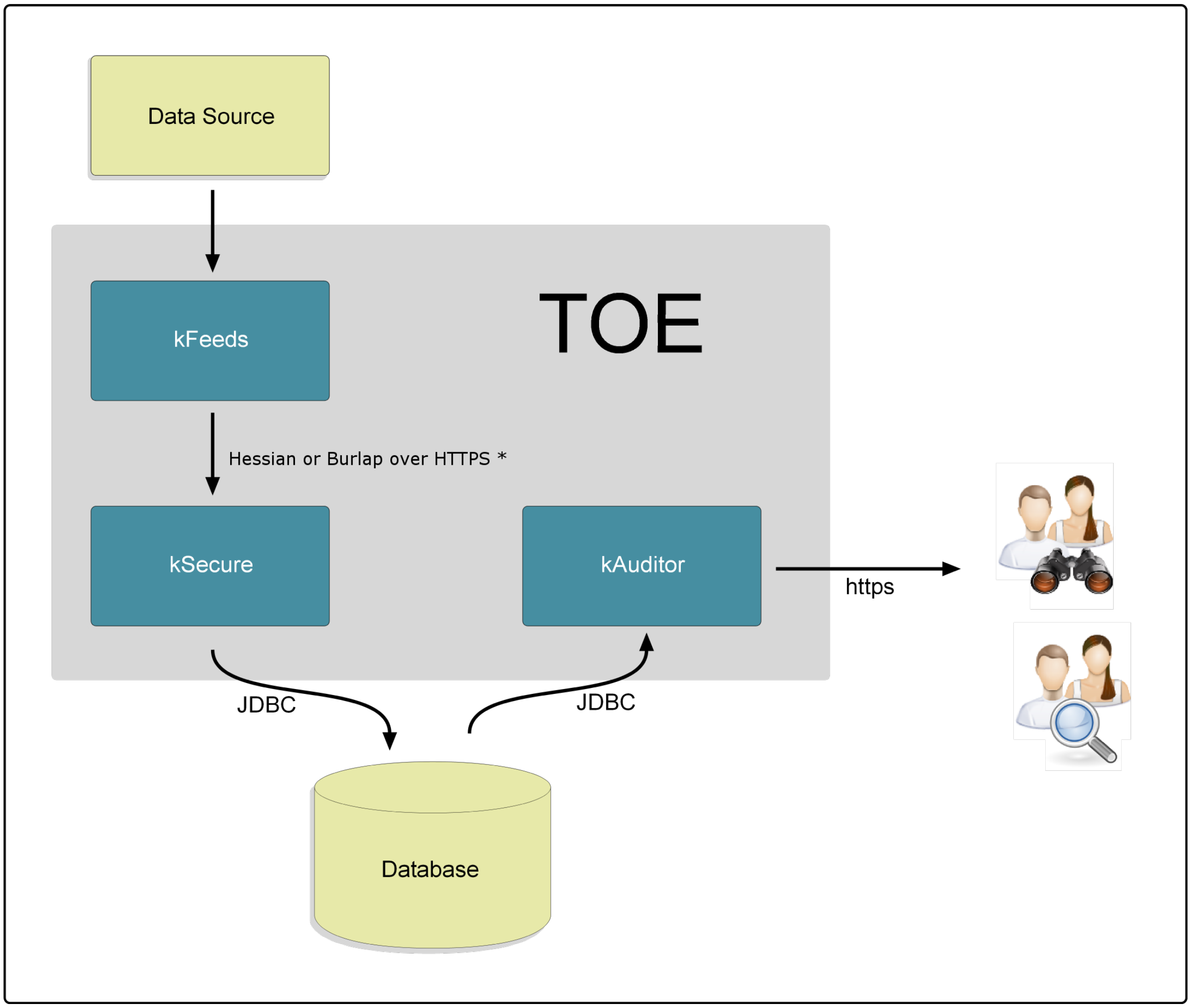
From a physical point of view, the recommended configuration involves several sub-systems.

1. At least one server to host kSecure, and typically more than one to achieve scalability and high availability.

2. At least one server to host kAuditor.

3. At least one server to host the database, and typically more than one to achieve high availability.

4. The kFeeds can be scattered across the organization, typically collocated with their corresponding data sources.



[*] HTTP and JMS protocols are out of the scope of the certification.

1.4.2. Security features

As mentioned above, the goal of the protocol is to add tamper-evidence properties to an incoming stream of records in real time at low computational cost and with integrity at the record level.

These objectives are accomplished by combining standard asymmetric-key cryptographic algorithms with chained hashing. Hash chaining is a standard cryptographic primitive based on computing one hash token per record. The hash token for a record is generated by computing the hash of the concatenation of the record with the hash of the previous record. Kinamik's protocol enhances this process by using public-key cryptography during the process to authenticate the chain.

SF.1 Using chained hashes allows the protocol to identify tampering on a record by record basis. Chained hashing is a well-known cryptographic primitive recognized by standards such as ANSI X9.95 and ISO/IEC 18014.

SF.2 Confidentiality, integrity and authenticity are accomplished at the data level. It is customary in many systems to accomplish those objectives by relying on access controls on the data. By contrast, once data has been processed by kSecure it has been rendered tamper evident, authentic and (optionally) confidential. Database access control is not needed to enforce neither of them.

SF.3 kAuditor can export data from the database and validate its integrity and authenticity.

SF.4 Authenticity is enforced through all the TOE data pipeline. The kFeeds authenticate to kSecure and push data over a secure channel. kSecure renders data authentic and tamper evident. kAuditor verifies data integrity and authenticity, and all data exported by kAuditor is digitally signed by kAuditor. The chain of custody is thus safeguarded.

1.4.3. Security attributes

- Role. This security attribute grants permissions to execute different operations. For this reason is used in the access control policies. Currently there are the following roles: View an audit trail, Auditing, Feed, Admin.

  The Auditing role allow users to execute the integrity verification operations.

  The Feed role allows users to send records. A user needs to be defined in the Feed when it is configured. Sometimes "kFeed" is used instead of "user" when the Feed role is mentioned. In those cases we are talking about the user defined in the kFeed.

  The view an audit trail role gives visibility on the audit trails. That implies that users will be able to execute on the visible audit trails the operations that provides the other roles assigned.

  The Admin role allows users to execute the admin operations.

  A user can have different roles at the same time

- Encryption flag. The encryption flag defines that the sequence of records stored in an audit trail is encrypted to provide confidentiality.

- Record position. This attribute defines the position of the record inside the sequence. It’s used to grant that the elements inside the sequence are consecutive and there are no gaps.

1.4.4. User data

- Sequence of records in transit to the TOE

Chapter 2. Conformance claims

This security target document is conformant with Common Criteria (CC) for Information Technology Security Evaluation version 3.1 Revision 3 and package EAL1 (Evaluation Assurance Level 1).

Security requirements are conformant with CC parts 2 and 3.

This security target document is not conformant with any Protection Profile (PP).

Chapter 3. Security objectives

This section deals only with the security objectives for the operational environment.

SO.1 All the computers hosting TOE subsystems must be hardened computers administered by competent and trusted personnel and protected from unauthorized physical modification.

SO.2 All the machines involved in the management of the cryptographic keys required by the TOE must be hardened machines administered by competent and trusted personnel and protected from unauthorized physical modification

Note there are no security requirements for the database environment. The underlying reason is that the purpose of TOE is to notarize data once it reaches the system so to make any future tampering evident. It is obvious of course that securing data storage is critical, but first that is outside the reach of the TOE and second that is the case where the TOE will be able to report the problem.

Chapter 4. Extended components definition

None.

Chapter 5. Security requirements

5.1. User data flow

5.1.1. FDP_ITC.1.kFeed Import of user data without security attributes

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_ITC.1.1 The TSF shall enforce the [assignment: none] when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.1.2 The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.

FDP_ITC.1.3 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: [assignment: none].

Application note:

- The communication between the source and the feed does not require any SPF. kFeed will accept everything that the source sends. Due to this neither the FDP_ACC.1 nor the FDP_IFC.1 dependencies are required.
- No security attributes are added to the data imported from the sources. That implies that dependence FMT_MSA.3 is not required.

5.1.2. FDP_ITT.3.kFeed-kSecure Integrity monitoring

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control, FDP_IFC.1 Subset information flow control, FDP_ITT.1 Basic internal transfer protection

FDP_ITT.3.1 The TSF shall enforce the [assignment: source access control SFP, record order flow control SFP] to monitor user data transmitted between physically-separated parts of the TOE for the following errors: [assignment: replay of any record, lose of any record in the complete sequence of records and unauthorized access].

FDP_ITT.3.2 Upon detection of a data integrity error, the TSF shall [assignment: show an error in the internal logs of kSecure and discard replayed records].

Application note:

- FDP_ACC.1 is satisfied by FDP_ACC.1.kFeed
- FDP_IFC.1 is satisfied by FDP_IFC.1.order
- FDP_ITT.1 is satisfied by FDP_ITT.1.kFeed-kSecure

5.1.3. FDP_ITT.1.kFeed-kSecure Basic internal transfer protection

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control, FDP_IFC.1 Subset information flow control

FDP_ITT.1.1 The TSF shall enforce the [assignment: source access control SFP, record order flow control SFP] to prevent the [selection: modification and **unauthorized access**] of user data when it is transmitted between physically-separated parts of the TOE.

Application note:

- About the selection:
  - **modification** for this SFR means: **replay of any record**, **lose of any record in the complete sequence of records**
  - **unauthorized access** is specific from our TOE.
- FDP_ACC.1 is satisfied by FDP_ACC.1.kFeed
- FDP_IFC.1 is satisfied by FDP_IFC.1.order

5.1.4. FDP_ACC.1.kFeed Subset access control

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1 The TSF shall enforce the [assignment: source access control SFP] on [assignment: **list of subjects:** kFeed, kSecure **objects:** sequence of records, kSecure audit trail id **operations**: kFeed sends a sequence of records to an audit trail of kSecure].

Application note:

- FDP_ACF.1 is satisfied by FDP_ACF.1.kFeed

5.1.5. FDP_ACF.1.kFeed Security attribute based access control

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control, FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1 The TSF shall enforce the [assignment: source access control SFP] to objects based on the following: [assignment: **list of subjects:** kFeed, kSecure **objects controlled:** sequence of records, kSecure audit trail id **SFP-relevant security attributes:** role].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [assignment: The kFeed has **feed** and **view audit trail (audit trail id)** roles].

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [assignment: none].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [assignment: none].

Application note:

- FDP_ACC.1 is satisfied by FDP_ACC.1.kFeed

5.1.6. FDP_IFC.1.order Subset information flow control

Hierarchical to: No other components.

Dependencies: FDP_IFF.1 Simple security attributes

FDP_IFC.1.1 The TSF shall enforce the [assignment: record order flow control SFP] on [assignment: **list of subjects:** kFeed, kSecure, **information:** sequence of records, kSecure audit trail id, user session **operations:** kFeed sends a sequence of records to an audit trail of kSecure].

Application note:

- FDP_IFF.1 is satisfied by FDP_IFF.1.order

5.1.7. FDP_IFF.1.order Simple security attributes

Hierarchical to: No other components.

Dependencies: FDP_IFC.1 Subset information flow control, FMT_MSA.3 Static attribute initialisation

FDP_IFF.1.1 The TSF shall enforce the [assignment: record order flow control SFP] based on the following types of subject and information security attributes: [assignment: **list of subjects:** kFeed, kSecure, **information controlled:** sequence of records, kSecure audit trail id, user session **security attributes:** record position].

FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: [assignment: record position is greater than or equal to one plus the position of the last record of the user session in the audit trail].

FDP_IFF.1.3 The TSF shall enforce the [assignment: none].

FDP_IFF.1.4 The TSF shall explicitly authorise an information flow based on the following rules: [assignment: none].

FDP_IFF.1.5 The TSF shall explicitly deny an information flow based on the following rules: [assignment: none].

Application note:

- FDP_IFC.1 is satisfied by FDP_IFC.1.order

5.1.8. FDP_DAU.1 Basic Data Authentication

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_DAU.1.1 The TSF shall provide a capability to generate evidence that can be used as a guarantee of the validity of [assignment: sequence of records].

FDP_DAU.1.2 The TSF shall provide [assignment: users with auditing role] with the ability to verify evidence of the validity of the indicated information.

5.1.9. FDP_UCT.1 Basic data exchange confidentiality

Hierarchical to: No other components.

Dependencies: FDP_IFC.1 Subset information flow control

FDP_UCT.1.1 The TSF shall enforce the [assignment: encryption information flow control SFP] to [selection: transmit] user data in a manner protected from unauthorised disclosure.

Application note:

- FDP_IFC.1 is satisfied by FDP_IFC.1.encr
- Dependencies FTP_ITC.1 and FTP_TRP.1 are not required because a trusted channel is not required by the Kinamik protocol , see Chapter 3, *Security objectives* section.

5.1.10. FDP_IFC.1.encr Subset information flow control

Hierarchical to: No other components.

Dependencies: FDP_IFF.1 Simple security attributes

FDP_IFC.1.1 The TSF shall enforce the [assignment: encryption information flow control SFP] on [assignment: **list of subjects:** kSecure, DB **information:** sequence of records, kSecure audit trail id **operations:** kSecure encrypts the sequence of records of a certain audit trail that are sent to DB].

Application note:

- FDP_IFF.1 is satisfied by FDP_IFF.1.encr

5.1.11. FDP_IFF.1.encr Simple security attributes

Hierarchical to: No other components.

Dependencies: FDP_IFC.1 Subset information flow control, FMT_MSA.3 Static attribute initialisation

FDP_IFF.1.1 The TSF shall enforce the [assignment: encryption information flow control SFP] based on the following types of subject and information security attributes: [assignment: **list of subjects:** kSecure, DB **information:** sequence of records, kSecure audit trail id **security attributes:** encryption flag].

FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: [assignment: the encryption flag of the kSecure audit trail is enabled].

FDP_IFF.1.3 The TSF shall enforce the [assignment: none].

FDP_IFF.1.4 The TSF shall explicitly authorise an information flow based on the following rules: [assignment: none].

FDP_IFF.1.5 The TSF shall explicitly deny an information flow based on the following rules: [assignment: none].

Application note:

- FDP_IFC.1 is satisfied by FDP_IFC.1.encr

5.1.12. FDP_ETC.2 Export of user data with security attributes

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_ETC.2.1 The TSF shall enforce the [assignment: none] when exporting user data, controlled under the SFP(s), outside of the TOE.

FDP_ETC.2.2 The TSF shall export the user data with the user data’s associated security attributes.

FDP_ETC.2.3 The TSF shall ensure that the security attributes, when exported outside the TOE, are unambiguously associated with the exported user data.

FDP_ETC.2.4 The TSF shall enforce the following rules when user data is exported from the TOE: [assignment: none].

Application note:

- Dependence FDP_IFC.1 and FDP_ACC.1 are not required because the authorization with the DB is outside the TOE.
- The integrity of the sequence of records that will exported is assured (FDP_SDI.2), also its authenticity (FDP_DAU.1) and optionally the sequence of records is encrypted (FDP_UCT.1)

5.1.13. FDP_SDI.2 Stored data integrity monitoring and action

Hierarchical to: FDP_SDI.1 Stored data integrity monitoring

Dependencies: No dependencies.

FDP_SDI.2.1 The TSF shall monitor user data stored in containers controlled by the TSF for [assignment: modification/deletion of any bit in a sequence of records, replay of any record, lose of any record in the complete sequence of records] on all objects, based on the following attributes: [assignment: sequence of records].

FDP_SDI.2.2 Upon detection of a data integrity error, the TSF shall [assignment: show the integrity error found in kAuditor].

5.1.14. FDP_ITC.2.kAuditor Import of user data with security attributes

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control

FDP_ITC.2.1 The TSF shall enforce the [assignment: auditor access control SFP] when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.2.2 The TSF shall use the security attributes associated with the imported user data.

FDP_ITC.2.3 The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

FDP_ITC.2.4 The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.

FDP_ITC.2.5 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: [assignment: none].

Application note:

- FDP_ACC.1 is satisfied by FDP_ACC.1.auditor
- Dependencies FTP_ITC.1, FTP_TRP.1 & FPT_TDC.1 do not apply because TOE does not require a secure channel to access to the DB (see Chapter 3, *Security objectives* section)

5.1.15. FDP_ACC.1.auditor Subset access control

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1 The TSF shall enforce the [assignment: auditor access control SFP] on [assignment: **list of subjects:** users, kAuditor **objects:** sequence of records, kSecure audit trail id **operations**: User executes the record integrity and record exportation on kAuditor].

Application note:

- FDP_ACF.1 is satisfied by FDP_ACF.1.auditor

5.1.16. FDP_ACF.1.auditor Security attribute based access control

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control, FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1 The TSF shall enforce the [assignment: auditor access control SFP] to objects based on the following: [assignment: **list of subjects:** users, kAuditor **objects controlled:** sequence of records, kSecure audit trail id **SFP-relevant security attributes:** role].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [assignment: only users with **auditing** and **view audit trail (audit trail id)** roles are allowed to execute the record integrity and record exportation operations on kAuditor].

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [assignment: none].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [assignment: none].

Application note:

- FDP_ACC.1 is satisfied by FDP_ACC.1.auditor

5.1.17. FDP_ACC.1.administrator Subset access control

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1 The TSF shall enforce the [assignment: administrator access control SPF] on [assignment: **list of subjects**: user **objects**: user **operations**: access to management tasks].

Application note:

- FDP_ACF.1 is satisfied by FDP_ACF.1.administrator

5.1.18. FDP_ACF.1.administrator Security attribute based access control

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control, FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1 The TSF shall enforce the [assignment: administrator access control SPF] to objects based on the following: [assignment: **list of subjects**: user **objects**: user **the SFP-relevant security attributes**: role].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [assignment: subject user has **administration** role].

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [assignment: none].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [assignment: none].

Application note:

- FDP_ACC.1 is satisfied by FDP_ACC.1.administrator

5.2. Management

5.2.1. FMT_MSA.3 Static attribute initialisation

Hierarchical to: No other components.

Dependencies: FMT_MSA.1 Management of security attributes, FMT_SMR.1 Security roles

FMT_MSA.3.1 The TSF shall enforce the [assignment: source access control SFP, auditor access control SFP, encryption information flow control SFP, record order flow control SFP, administrator access control SPF] to provide [selection, choose one of: restrictive] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [assignment: administrator] to specify alternative initial values to override the default values when an object or information is created.

5.2.2. FMT_MSA.1 Management of security attributes

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control,FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions,

FMT_MSA.1.1 The TSF shall enforce the [assignment: source access control SFP, auditor access control SFP, encryption information flow control SFP, record order flow control SFP, administrator access control SPF] to restrict the ability to [selection: query, modify] the security attributes [assignment: role, encryption flag, record position] to [assignment: administrator].

Application Note:

- FDP_ACC.1 is satisfied by FDP_ACC.1.administrator
- **Modify** ability only applies to role

5.2.3. FMT_SMR.1 Security roles

Hierarchical to: No other components.

Dependencies: FIA_UID.2 User identification before any action

FMT_SMR.1.1 The TSF shall maintain the roles [assignment: administrator, auditor, feed, view audit trail (id)].

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

Application note:

- There is one **view audit trail** role for each audit trail defined in the system. The id of the audit trail is used to identify the role. That is the reason this role appears in the ST as **view audit trail (id)**

5.2.4. FIA_UID.2 User identification before any action

Hierarchical to: FIA_UID.1 Timing of identification

Dependencies: No dependencies.

FIA_UID.2.1 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

5.2.5. FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components.

Dependencies: No dependencies.

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: [assignment: query roles granted to a user, modify roles granted to a user, query encryption flag, create audit trail, query record position].

Application note:

- The record position is defined automatically in the kFeed. There is no management function accessible to users to set this attribute

5.2.6. FIA_UAU.2 User authentication before any action

Hierarchical to: FIA_UAU.1 Timing of authentication

Dependencies: FIA_UID.2 Timing of identification

FIA_UAU.2.1 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Chapter 6. TOE summary specification

This section presents a description of how the TOE SFRs are satisfied.

6.1. Role management, access control and information flow

Some notes about how TOE has satisfied the role management defined in the SFR. The concept **roles** defined in the SFRs is equivalent to **permissions** in TOE. A **role** in TOE is a group of users that share the same **permissions**. The TOE **roles** are assigned to audit trails to provide visibility to the users that belongs to the roles and apply the auditing and feed **permissions** to these audit trails.

FIA_UAU.2, FIA_UID.2, FDP_ACC.1.administrator, FDP_ACF.1.administrator, FMT_MSA.1

TOE provides a GUI to users to log in kAuditor. TOE checks the permissions associated to the user and it will show the management functions only if the user is authorized (User has **Permit Admin** permissions)

FMT_MSA.3

The TOE allows to the administrators overwrite the initial values of the roles. By default a role has not users assigned to it, it has not any permission and it is not assigned to any audit trail.

The encryption flag is set during the creation of the audit trail. TOE does not allow the update of the encryption flag once it's set.

The record position is set automatically by the kFeed. It can't be updated.

FMT_SMR.1

The kAuditor GUI allows associating the roles defined in the SFR to users (to grant to users administrator, auditor and feed permissions) and also it allows assign the roles to the audit trails to provide users assigned to the roles the view audit trail permission.

FMT_SMF.1

Because the role management defined in the SFRs has a special implementation in the TOE the list of management functions that the GUI kAuditor provides is a little bit different:

- **Create role**. kAuditor GUI provides a functionality to create a new role and assign users and permissions to it.
- **Query list of roles**. kAuditor GUI provides a functionality to list all TOE roles.
- **Query role**. kAuditor GUI provides a functionality to view the details of a certain role (users and permissions)
- **Modify role**. kAuditor GUI provides a functionality to modify the details of a role (users assigned to role, and permissions)
- **Create audit trail**. kAuditor GUI provides a functionality to create a new audit trail and assign roles to it.
- **Query list of audit trails**. kAuditor GUI provides a functionality to list all TOE audit trails.

- **Query audit trail**. kAuditor GUI provides a functionality to list all the details of an audit trail (like the roles assigned to the audit trail and the encryption flag)
- **Modify audit trail**. kAuditor GUI provides a functionality to modify the details of an audit trail (Like the roles assigned to the audit trail)

Query record position is not an admin functionality. kFeeds assign automatically the position and it can not be changed. kAuditor GUI shows the record position when it exports and shows data (see FDP_SDI.2)+

kAuditor does not provide any functionality to implement **Query permissions granted to a user** and **Modify permissions granted to a user** directly.

For the first case we can use **Query list of roles** to list all roles and **Query role** in each role to check if the user belongs the role and track the permissions. Also the **Query list of audit trails** and **Query audit trail** must be used to check the view audit trail permission of the user.

For the second case we can **Query list of roles** to list all roles and **Modify role** in each role to modify the permissions in the roles where user belongs. Also the **Query list of audit trails** and **modify audit trail** must be used to change the view audit trail permission of the user.

6.2. Import of the user data to TOE

FDP_ITC.1.kFeed

The kFeeds are the TOE components that receive the sequence of records that a source sends. Currently, following record data types generated by the source are supported: log4j, plain text, syslog, Oracle® WebLogic audit events, Sun® JDBC™ SQL statements, Sun® OpenSolaris™ audit events.

FDP_ITT.3.kFeed-kSecure,
FDP_ITT.1.kFeed-kSecure

kFeeds sends to kSecure a ordered sequence of records. kSecure checks that the records contained in each sequence of records are consecutive. If they are not and the position of the disordered record is lower or equal than the previous one, the record is discarded because it is assumed as duplicated. If the position is greater than the previous one + 1, the records is accepted. In both cases an alert is throw.

FIA_UAU.2, FIA_UID.2,
FDP_ACC.1.kFeed,
FDP_ACF.1.kFeed

The kFeed has a properties file that contains its credentials. kSecure will use these credentials to see which privileges has associated to the kFeed and validate if it is authorized to store records in an audit trail (the kFeed has **Permit Feed** privileges and it is assigned to the audit trail).

FDP_IFC.1.order,
FDP_IFF.1.order

kSecure only accepts records which position is equal or greater than one plus the position of the last record of the user session in the audit trail.

6.3. Storage of user data

FDP_DAU.1, FDP_UCT.1, FDP_SDI.2, FDP_ETC.2

kSecure applies a cryptographic protocol for electronic signature to all the sequences of records received from kFeeds (FDP_ITT.1.kFeed-kSecure), and saves the output into the database (FDP_ETC.2). This protocol uses chain hashing to provide integrity at a granular level (FDP_SDI.2), and digital signature for authenticity (FDP_DAU.1).

The kSecure sub-system will always encrypt records sent to audit trails that have the encryption policy activated (FDP_UCT.1). Note that confidential audit trails are thus encrypted before being stored at the database, so confidentiality is enforced at the data level.

FDP_IFC.1.encr, FDP_IFF.1.encr

kSecure validates that the audit trail associated to the sequence of records has the encrypted flag enabled.

6.4. Integrity verification

FDP_ITC.2.kAuditor

kAuditor reads from the DB the chains stored there by kSecure.

FDP_SDI.2

Integrity verification is performed per user request using the kAuditor GUI or when showing and exporting data. This verification is performed by re-computing the hash chain of the stored records and comparing it with the stored hash chain. kAuditor also verifies that the hash chain has been digitally signed by kSecure.

FIA_UAU.2, FIA_UID.2, FDP_ACC.1.auditor, FDP_ACF.1.auditor

kAuditor provides a GUI to users to log in kAuditor. kAuditor checks the privileges associated to the user and it will show only the authorized audit trails for auditing the integrity (TOE roles are assigned to the audit trail and have **Permit Audit** permissions).