



Audit Trail and Node Authentication

Table of Contents

Table of Contents	1
1. INTRODUCTION	2
2. OBJECTIVES	2
3. SCOPE	2
4. PROFILE DETAILED DESCRIPTION	3
4.2. Roles Involved	3
4.3. Actor Diagram	3
4.4. Transaction Information	4
4.4.1. Record Audit Event	4
4.4.1.1. Description	4
4.4.1.2. Interaction Diagram	4
4.4.1.3. Record Audit Event Message	5
4.4.1.3.1. Message Semantics	5
4.4.1.3.2. Protocol Requirements	7
4.4.2. Authenticate Node	7
4.4.2.1. Description	7
4.4.2.2. Interaction Diagram	7
4.4.2.3. Authenticate Node Message	8
4.4.2.3.1. Message Semantics	8
4.4.2.3.2. Protocol Requirements	8

1. INTRODUCTION

Audit Trail and Node Authentication profile describes the security environment assumed for the node such as user identification, authentication, authorization, access control etc. It establishes security measures to provide information confidentiality, data integrity and user accountability.

2. OBJECTIVES

The objectives of implementing this profile are:

1. Providing user accountability by auditing activities to assess compliance with a secure domain's policies.
2. Providing information confidentiality by defining bi-directional certificate based node authentication for the communications of the nodes.
3. Limiting network access between nodes.
4. Limiting access to each node to authorized users.

3. SCOPE

This profile supports a process of auditing of events and message level security. It is intended to identify nodes with known security characteristics as well as track security events. The intended scope for this profile includes:

- Before, during and after emergency situations (where emergency systems are in use)
- Among warning systems, emergency information systems, sensor systems, dissemination systems and internal systems.

The transactions specified in this profile are intended to be exchanged between emergency organizations' systems. It is expected that emergency systems have had necessary infrastructure, which is specified in the profile, to send and receive the messages in a secured way, using an agreed syntax.

4. PROFILE DETAILED DESCRIPTION

4.2. Roles Involved

Actor	Description
Local Secure Node	The Secure Node is the legal organization who is responsible for providing security audit logging and reasonable access controls.
Audit Record Repository	The Audit Record Repository is the database where audit messages are stored.
Remote Secure Node	The Secure Node is the legal organization who is responsible for providing security audit logging and reasonable access controls.

4.3. Actor Diagram

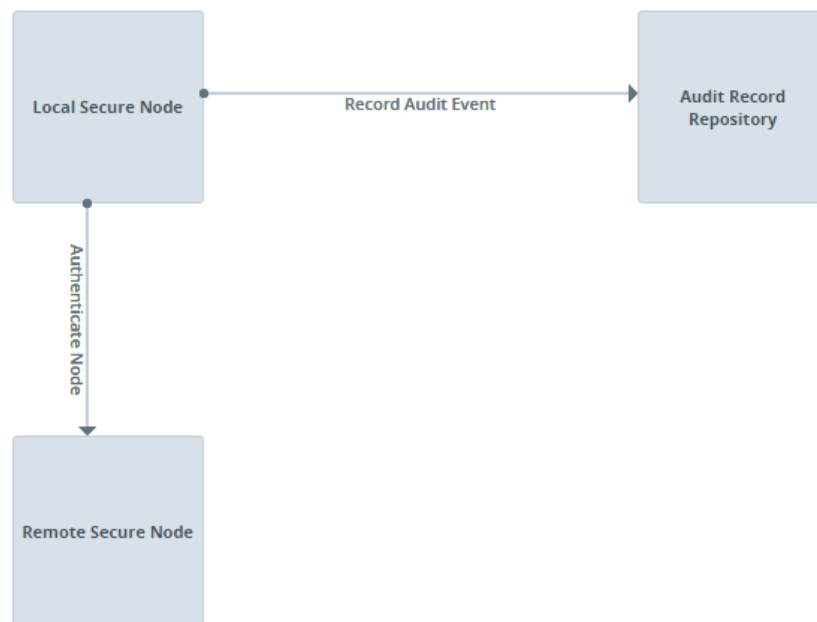


Figure above shows the actors directly involved in the Audit Trail and Node Authentication Profile and the possible messages to be exchanged between them.

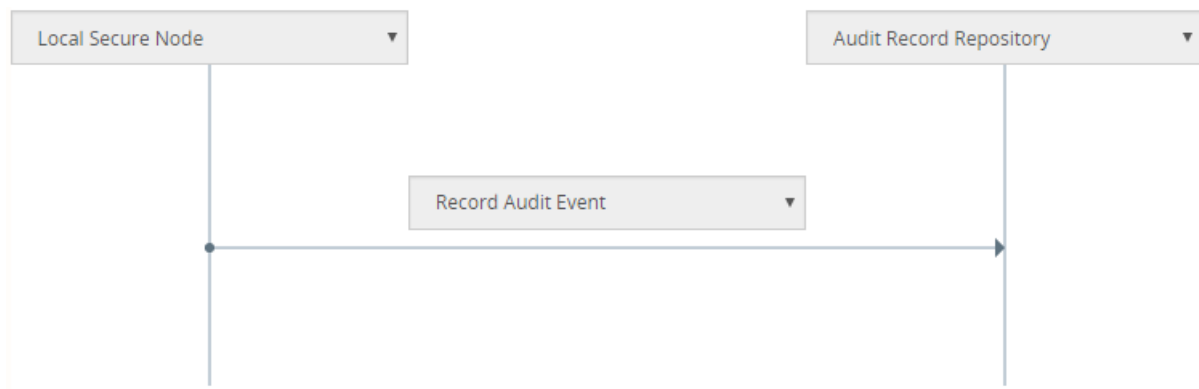
4.4. Transaction Information

4.4.1. Record Audit Event

4.4.1.1. Description

Categories	Description and Values
Identifier	1459164153379
Description	An Audit Log is a record of actions performed by users or systems. Audit Record is created when a C2-SENSE profile transaction-related event occurs or when a non-transaction event occurs. Which kind of actions are recorded or the usage of audit records by Audit Record Repository is beyond the scope of this profile.
Actors	Local Secure Node, Audit Record Repository
Pre-conditions	The event that requires the creation of an audit message is triggered.
Post-conditions	Audit Log is sent and saved in Audit Record Repository.

4.4.1.2. Interaction Diagram



4.4.1.3. Record Audit Event Message

4.4.1.3.1. Message Semantics

```
<?xml version="1.0" encoding="UTF-8"?>
<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" type="ActiveParticipantType"/>
        <xs:element name="AuditSourceIdentification" type="AuditSourceIdentificationType"/>
        <xs:element name="ParticipantObjectIdentification"
type="ParticipantObjectIdentificationType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="EventIdentificationType">
    <xs:sequence>
      <xs:attribute name="EventOutcomeIndicator">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:enumeration value="0"/>
            <xs:enumeration value="4"/>
            <xs:enumeration value="8"/>
            <xs:enumeration value="12"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:element name="EventDateTime" type="xs:date"/>
      <xs:element name="EventActionCode">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="C"/>
            <xs:enumeration value="R"/>
            <xs:enumeration value="U"/>
            <xs:enumeration value="D"/>
            <xs:enumeration value="E"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="EventID">
        <xs:complexType>
          <xs:sequence>
            <xs:attribute name="codeSystemName" type="xs:string"/>
            <xs:element name="displayName" type="xs:string"/>
            <xs:element name="code" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```

        </xs:complexType>
    </xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="ActiveParticipantType">
    <xs:sequence>
        <xs:attribute name="UserIsRequestor" type="xs:boolean"/>
        <xs:element name="userName" type="xs:string"/>
        <xs:element name="UserID" type="xs:string"/>
        <xs:element name="NetworkAccessPointTypeCode">
            <xs:simpleType>
                <xs:restriction base="xs:integer">
                    <xs:pattern value="[1-5]"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="NetworkAccessPointID" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="AuditSourceIdentificationType">
    <xs:sequence>
        <xs:attribute name="AuditSourceID" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ParticipantObjectIdentificationType">
    <xs:sequence>
        <xs:element name="ParticipantObjectTypeCode">
            <xs:simpleType>
                <xs:restriction base="xs:integer">
                    <xs:pattern value="[1-4]"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="ParticipantObjectID" type="xs:string"/>
        <xs:element name="ParticipantObjectIDTypeCode">
            <xs:complexType>
                <xs:sequence>
                    <xs:attribute name="codeSystemName" type="xs:string"/>
                    <xs:element name="displayName" type="xs:string"/>
                    <xs:element name="code" type="xs:string"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:element name="ParticipantObjectDetail">
            <xs:complexType>
                <xs:sequence>
                    <xs:attribute name="value" type="xs:string"/>
                    <xs:element name="type" type="xs:string"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>

```

```

        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:schema>

```

4.4.1.3.2. Protocol Requirements

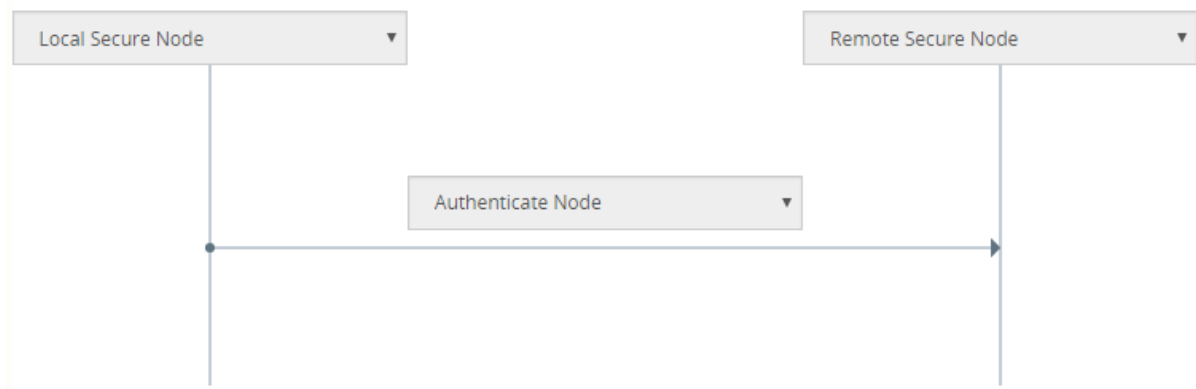
Use REST protocol and handle it through POST request.

4.4.2. Authenticate Node

4.4.2.1. Description

Categories	Description and Values
Identifier	1459164153440
Description	The Local Secure Node presents its identity to a Remote Secure Node, and authenticates the identity of remote node. After this mutual authentication, other secure transactions may take place through this secure pipe between two nodes.
Actors	Local Secure Node, Remote Secure Node
Pre-conditions	The user managing the Secure Node is already authenticated to the system, which can be done through Enterprise User Authentication profile, but not necessarily.
Post-conditions	The Local Secure Node either authenticates the Remote Secure Node by performing certificate validation based on signature of a trusted CA; or rejects the communication when the certificate validation fails.

4.4.2.2. Interaction Diagram



4.4.2.3. Authenticate Node Message

4.4.2.3.1. Message Semantics

Through certificates.

4.4.2.3.2. Protocol Requirements

Use REST protocol and handle it through POST request.