eIDAS-Node Error and Event Logging

Version 1.4.4

Document history

Version	Date	Modification reason	Modified by
1.0	06/10/2017	Extracted from Installation Manual	DIGIT
1.4.1	15/06/2018	Reuse of document policy updated. Version changed to match the corresponding Release.	DIGIT
1.4.4	14/12/2018	To reflect updates in the new release such as those relating to message logging.	DIGIT

Disclaimer

This document is for informational purposes only and the Commission cannot be held responsible for any use which may be made of the information contained therein. References to legal acts or documentation of the European Union (EU) cannot be perceived as amending legislation in force or other EU documentation.

The document contains a brief overview of technical nature and is not supplementing or amending terms and conditions of any procurement procedure; therefore, no compensation claim can be based on the contents of the present document.

Table of contents

List of figures

List of tables

1. Introduction

This document is intended for a technical audience consisting of developers, administrators and those requiring detailed technical information on how to configure, build and deploy the eIDAS-Node application.

This document provides information on the eID implementation of error and event logging as a building block for generating an audit trail of activity on the eIDAS Network. It describes the files that are generated, the file format, the components that are monitored and the events that are recorded. It is also presents points that should be considered when planning, implementing and operating an auditing environment.

1.1. Purpose

The purpose of this document is to describe the strategy for logging of errors and events in the eIDAS-Node.

1.2. Document Structure

This document is divided into the following sections:

- Chapter 1 – *Introduction*: this section.
- Chapter 2 – *What is an audit trail?* Introduces the concept of an audit trail and discusses the forensics and diagnosis of audit trail analysis.
- Chapter 3 – *Standards, frameworks and best practice* provides a brief introduction to industry standards and guidelines that should be considered when implementing a logging and auditing scheme.
- Chapter 4 – *Log generation* shows the format and describes the information attributes that are contained in a log record.
- Chapter 5 – *Logging configuration settings* discusses log management requirements and recommendations.
- Chapter 6 – *Log files* describes the locally stored log files and presents a matrix of which events are logged, in which log file and which response would be triggered.
- Chapter 7 – *Operational considerations* contains information that should be taken into account when implementing a logging process.
- Chapter 8 – *References* contains a list of reference material with links.
- Appendix A – *eIDAS-Node Error Codes and Error Messages* contains a list of error code and associated message properties.

1.3. Other technical reference documentation

We recommend that you also familiarise yourself with the following eID technical reference documents which are available on **CEF Digital Home > eID > All eID services > eIDAS Node integration package > View current version > Useful documentation**

- *eIDAS-Node Installation, Configuration and Integration Quick Start Guide* describes how to quickly install a Service Provider, eIDAS-Node Connector, eIDAS-Node Proxy Service and IdP from the distributions in the release package.

The distributions provide preconfigured eIDAS-Node modules for running on each of the supported application servers.

- *eIDAS-Node Installation and Configuration Guide* describes the steps involved when implementing a Basic Setup and goes on to provide detailed information required for customisation and deployment.
- *eIDAS-Node National IdP and SP Integration Guide* provides guidance by recommending one way in which eID can be integrated into your national eID infrastructure.
- *eIDAS-Node Demo Tools Installation and Configuration Guide* describes the installation and configuration settings for Demo Tools (SP and IdP) supplied with the package for basic testing.
- *eIDAS-Node and SAML* describes the W3C recommendations and how SAML XML encryption is implemented and integrated in eID. Encryption of the sensitive data carried in SAML 2.0 Requests and Assertions is discussed alongside the use of AEAD algorithms as essential building blocks.
- *eIDAS-Node Security Considerations* describes the security considerations that should be taken into account when implementing and operating your eIDAS-Node scheme.
- *eIDAS-Node Error Codes* contains tables showing the error codes that could be generated by components along with a description of the error, specific behaviour and, where relevant, possible operator actions to remedy the error.

2. What is an audit trail?

A log is a record of the events occurring within an organisation's systems and networks. Logs are composed of log entries; each entry contains information related to a specific event that has occurred within a system or network.

An audit trail consists of a number of log records providing documentary evidence of the sequence of activities that have affected at any time a specific operation, procedure or event.

In a stand-alone system all the security [1] relevant activity in individual components can be isolated, thus allowing us to maintain a time-ordered list of all actions and events (e.g. audit trail) that happened in the system and enabling us to audit the system by simply following the information trail in a single place, obeying to a defined notation and in a specific timeline.

The number, volume and variety of different systems operating in an eIDAS-Network's distributed architecture has created the need for security log management and safeguards to help protect the confidentiality, integrity, and availability of service. With this in mind an audit trail must meet two very important requirements:

- The possibility of reconstructing an entire transaction by linking all related request/response identifiers throughout the complete path from Service Provider to Identity Provider and back; starting from the end.
- The correlation between an incoming request and (all related) outgoing request(s), as well as the corresponding replies. More specifically, the eIDAS specifications [2] require the nodes to store the following elements:
  - node's identification;
  - message identification;
  - message date and time.

In order to meet these requirements the eIDAS-Node implementation is required to record and retain audit-logging [3] information sufficient to answer the following questions:

- Who performed the logging (the main component or an external module/entity)?
- What activity was performed? (e.g. Input, Process, Output)
- Who or what performed the activity, including where or on what system the activity was performed from (Subject)?
- What the activity was performed on (Object)?
- When was the activity performed?
- What tool(s) was used to perform the activity?
- What was the status (such as success/failure), outcome, or result of the activity?

2.1. Forensics and diagnostics

Analysis of the audit trail will help to identify use and abuse of the eIDAS Network, including the following conditions:

- Sequencing failure
- Excessive use
- Data changes
- Fraud and other criminal activities
- Suspicious, unacceptable or unexpected behavior
- Modifications to configuration
- Application code file and/or memory changes input validation failures e.g. protocol violations, unacceptable encodings, invalid parameter names and values
- Output validation failures e.g. database record set mismatch, invalid data encoding
- Authentication successes and failures
- Authorisation (access control) failures
- Session management failures e.g. cookie session identification value modification
- Application errors and system events e.g. syntax and runtime errors, connectivity problems, performance issues, third party service error messages, file system errors, file upload virus detection, configuration changes
- Application and related systems start-ups and shut-downs, and logging initialisation (starting, stopping or pausing)
- Use of higher-risk functionality e.g. network connections, addition or deletion of users, changes to privileges, assigning users to tokens, adding or deleting tokens, use of systems administrative privileges, access by application administrators, all actions by users with administrative privileges, access to payment cardholder data, use of data encrypting keys, key changes, creation and deletion of system-level objects, data import and export including screen-based reports, submission of user-generated content - especially file uploads
- Legal and other opt-ins e.g. permissions for mobile phone capabilities, terms of use, terms & conditions, personal data usage consent, permission to receive marketing communications

3. Standards, frameworks and best practice

Standards and frameworks like COBIT [4] (Control Objectives for Information and Related Technology) by the Technology IT Governance Institute and the Information Systems Audit and Control Association (ISACA) and ITIL (Information Technology Infrastructure Library) provide information on IT governance and auditing. However there is little regulatory guidance on the specific controls, the standard of due care means doing at least what one's peers are doing as well as following security best practices frameworks as proposed by COBIT, BITS, COSO and standards including ISO 17799 and NIST SP 800-53 and ISF Standard of Good Practice, ISO / ISO 27002 [5].

Note: The eIDAS-Node logs may contain person identification data, hence these logs should be handled and protected appropriately in accordance with the European privacy regulations [Dir. 95/46/EC] and [Reg. 2016/679].

[Reg. 2016/679] REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC.

[Dir. 95/46/EC] Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data.

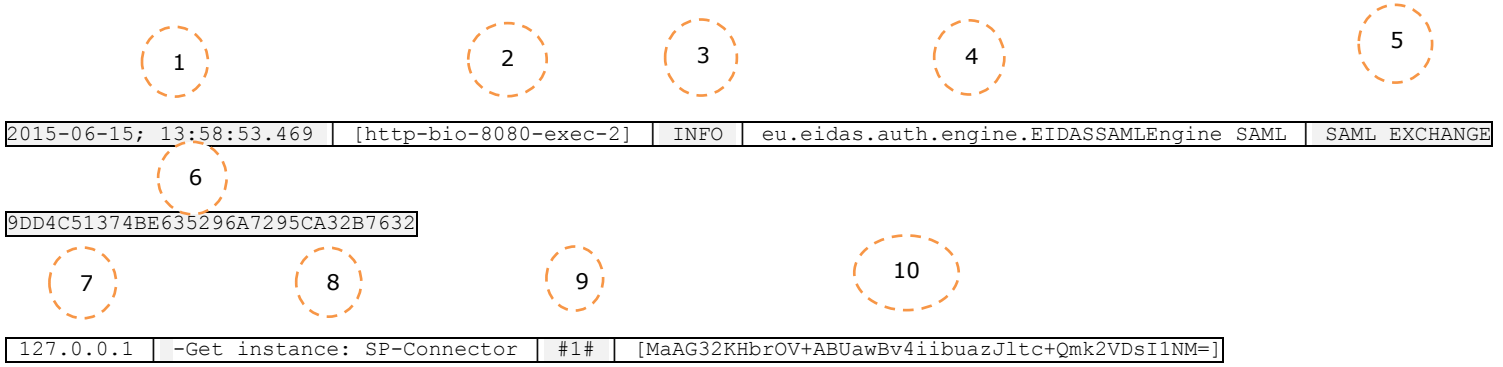
4. Log generation

The following table shows the information attributes that are contained in a log record.

Table 1: Log record attributes

Name	See Figure 1	Description
Date time	1	The date time of the of the logging event
Thread	2	Outputs the name of the thread that generated the logging event.
Level	3	Outputs the level of the logging event. • **ERROR**: some type of failure has occurred, users are probably being affected (or will soon be) and the fix probably requires human intervention. • **WARN**: an unexpected technical or business event occurred, users may be affected, but probably no immediate human intervention is required. Support personnel will want to review these issues a soon as possible to understand what the impact is. Basically any issue that needs to be tracked but may not require immediate intervention. • **INFO**: things we want to see at high volume in case we need to forensically analyse an issue. Typical business exceptions can go here (e.g. login failed due to bad credentials). • **DEBUG**: used for entry/exit of most non-trivial methods and detailed information on the flow through the system. This will be written to logs only. • **TRACE**: this would be for extremely detailed and potentially high volume logs that you do not typically want enabled even during normal development.
ClassName	4	Outputs location information of the caller which generated the logging event.
Event	5	If available, the type of logging event.
sessionId	6	If available the session Id.
IpAddress	7	If available the Remote IP address.
Message	8	The content of the message
EventNumber	9	Sequential increasing number of the event
Hash	10	A hash generated of the line log content guaranteeing integrity Uses SHA-256 (see details in $4.2).

The following is an example of log content showing three log records.




```
2015-06-15; 13:58:57.825 [http-bio-8080-exec-2] DEBUG eu.eidas.auth.engine.EIDASSAMLEngine SAML_EXCHANGE -
9DD4C51374BE635296A7295CA32B7632 -127.0.0.1 -null #2# [/d5F9UqFZy3OY37PA500TGpepPczdUzbJigiZxGIa2E=]


2015-06-15; 13:59:07.212 [http-bio-8080-exec-2] INFO  eu.eidas.node.auth.connector.AUCONNETORSAML SAML_EXCHANGE -
9DD4C51374BE635296A7295CA32B7632 -127.0.0.1 -Connector - Processing SAML Request with ID _a43526c05fc8168879b5789ba0d62744 #3#
[AeRp3ofGkl3gcIEH4QypViREHPB9Kw+hFFpt1DdUKGc=]
```

Figure 1: Example of log content

Notes: The following shows the attribute delimiters and their meanings:

is used as a field delimiter

Important: Additional attributes may be added in future phases of this project.

Figure 2 and Figure 3 below show examples of log records on a Service Provider and on an eIDAS-Node.

```
1#13Feb200918:22:59#Auth#POLITO#201.202.203.204#IT-eIDASNode#193.194.195.196#12345#
654ACEFD#
……
25#13Feb200918:23:21#Auth#IT-
eIDASNode#193.194.195.196#POLITO#201.202.203.204#34567#
ABE5737D#POLITO#12345
……
```

Figure 2: Log records on a Service Provider

```
2015-06-15; 13:58:53.469 [http-bio-8080-exec-2] INFO
eu.eidas.auth.engine.EIDASSAMLEngine SAML_EXCHANGE -
9DD4C51374BE635296A7295CA32B7632 -127.0.0.1 -Get instance: SP-Connector  #1#
[MaAG32KHbrOV+ABUawBv4iibuazJltc+Qmk2VDsI1NM=]
2015-06-15; 13:59:07.212 [http-bio-8080-exec-2] INFO
eu.eidas.node.auth.connector.AUCONNECTORSAML SAML_EXCHANGE -
9DD4C51374BE635296A7295CA32B7632 -127.0.0.1 -Connector - Processing SAML Request
with ID _a43526c05fc8168879b5789ba0d62744 #3#
[AeRp3ofGkl3gcIEH4QypViREHPB9Kw+hFFpt1DdUKGc=]
...
2015-06-15; 13:59:07.393 [http-bio-8080-exec-2] INFO
eu.eidas.auth.engine.EIDASSAMLEngine SAML_EXCHANGE -
9DD4C51374BE635296A7295CA32B7632 -127.0.0.1 -Get instance: Connector-Service  #15#
[lJCReY/HA7Kcc9HyKuJww2qFRvjSxkQfmw/1sBGnxH8=]
2015-06-15; 13:59:07.529 [http-bio-8080-exec-2] INFO
eu.eidas.node.auth.connector.AUCONNECTORSAML SAML_EXCHANGE -
9DD4C51374BE635296A7295CA32B7632 -127.0.0.1 -Connector - Processing SAML Request
with ID _a43526c05fc8168879b5789ba0d62744 #16#
[zsSBlcAwF8Lg1Ig7u2S//OtoklvnnLZnAPcWX6Quci8=]
...
2015-06-15; 13:59:08.445 [http-bio-8080-exec-2] INFO
eu.eidas.auth.engine.EIDASSAMLEngine SAML_EXCHANGE -
9DD4C51374BE635296A7295CA32B7632 -127.0.0.1 -Get instance: Service  #29#
[/K59pCfDIfHDTBiQKc1n5VLKBtdOprWsEumnndh8TIQ=]
2015-06-15; 13:59:08.697 [http-bio-8080-exec-2] INFO
eu.eidas.node.auth.service.AUSERVICESAML SAML_EXCHANGE -
9DD4C51374BE635296A7295CA32B7632 -127.0.0.1 -Service - Processing SAML Request with
ID _d41ae1dc063c7e4a87f17e57fdbc1329 #31#
[rkLbukPhl+5ia+wzzJyVY2DHHfz2JlXPvpT1CmHRZgw=]
```

Figure 3: Log records on IT-eIDAS-Node

4.1. SAML message logging

The meaning of all the logging entries corresponding to logging of SAML messages are described in the table below.

Entry	Description	Remarks
Timestamp	Current time	

OpType	*Operation type can be receives, generates, or sends.*	
Origin	The address of the previous web page or component from which a link to the currently requested page was followed. (i.e., HTTP _referer_header.)	
Destination	The service or component this message is targeted to.	
samlHash	The hash of the SAML message.	Uses SHA512 (see details in $4.2).
originatingMsgId	The Id of the previous message in the communication flow. This field has a value only for generated messages.	N/A except for OpType "generates" types
msgId	The Id of this message.	
NodeId	The Id of the node or component from which this message was received (in case of OpType *receives*) or to which it will be sent (in case of OpType *generates* and *sends*). For instance, the Id of of eIDAS Nodes will be the *EntityId* from node's SAML Metadata.	N/A for generates, .
inResponseTo	The Id of the message which correlates a response message with a request message.	This field is present only for logging of responses
statusCode	SAML StatusCode coming with the SAML Response.	This field is present only for logging of responses

SAML messages are logged in different logging points in the eIDAS-Node. A few examples are provided bellow.

The correlation of requests and responses can be done using the entries: *msgId*, *originatingMsgId*, and *inResponseTo*.

For more details on the logging points and an example of an end-to-end logging trail SP (country A) ↔ ... ↔ Idp (country B) for the SAML messages, please refer to *Message logging improvements in eIDAS-Node v1.4.4* → https://ec.europa.eu/cefdigital/wiki/x/Ex5HB [1]

[1] The URLs with ec.europa.eu domain in this document are only accessible to the experts registered with the eIDAS eID Implementation community.

Log records on eIDAS Service

SP to eIDAS Connector

```
Timestamp     2018-12-04 15:34:36.492, #34# [2AlYAcjNdU66gK9f6M298IgI7Pqv80sE4ByBJGekqf0=]
OpType        eIDAS Connector receives request from SP, #35# [ocKOfsVgASLDNjUPquSLrjeZZpsWBWcTvhqFW17eNOY=]
NodeId        http://localhost:8080/SP/metadata, #36# [riL+AzeHBanElbpsWYzxQlJzaLIUfJDtUz1ZsVWegdw=]
Origin        http://localhost:8080/SP/IndexPage.action, #37# [793phSM78CM9ZZiVI6GyVAJmdgo91//TNlVvQ5K1FAM=]
Destination   http://localhost:8080/EidasNode/ServiceProvider, #38# [KqMG9LK+YhbTWYILOTwoS5QdvKjiEugGVwtNpgDes00=]
samlHash      Nu84d0ArwihNCovmIGg60Q4a25ayqzTCDVsG74yN7FqgX9asPiSClZyxsdweaOlBwnBMYQZI+qKUHLE9DBSwqQ==, #39#
[ZVzCEGdChwdw+HKWeu1u6Ek5mFchVwxAMs38eKw0mZo=]
originatingMsgId N/A, #40# [QEnIRYRbvvI4PeRbl5MtOmUn1x1x3j9LVRLf9naEPug=]
msgId         _d7GC643-eO9sCdegVOjEcXQNNk2ki6sNk2C7CNWsEi.SDNipIE2kwtU3d.k04Ht #41#
[z9zJU14meXJrb72SYjQClcoEDptblheWUcs0T8pE3oI=]



Timestamp     2018-12-04 15:34:36.619, #65# [ZMkZZv4QiiRyM+BQm/EJOIHeBw+0J+cOqVDzHDQPS/Q=]
OpType        eIDAS Specific Connector generates request to eIDAS Connector, #66# [+ZggT6KAo+JrrxZOAAyTJ2jz9KCuSEzSJJdNn7tWIKM=]
NodeId        N/A, #67# [lG/WoumAKC+QHDkNk4MvIH/guxIrDvnNqO6QD3lELnk=]
Origin        N/A, #68# [5uY5IHF6qgp527JcpdBtpsxUOGojCrvMdaWxG2PaSRs=]
Destination   N/A, #69# [Csg0I+/ZbGmNsvOU+U601zK45sfZxRHZxyAAVxCp8Pw=]
samlHash      2ULtCIY/MSr4PjM78/+Oy17Ax1KP7ZRRT6ZK3PiSJDT+rdEklyqlztT/z5nrV1s9l8qC8c5PduxwVX+3T8sAgg==, #70#
[6NIK4I/mRgRHIyCrL3+z359hMogMDt6cZmrgjl6sMxw=]
originatingMsgId _d7GC643-eO9sCdegVOjEcXQNNk2ki6sNk2C7CNWsEi.SDNipIE2kwtU3d.k04Ht, #71#
[iJB3eT/A/jiqYq7WXO6NTgh7clgbiOlA2DH7fr+CknA=]
msgId         _5HzGYnqYAu8IVoEHAyxfSSQkGJb4HsUkuz6ZiggB2G.yhV4I_gZGOPIHyNb64D7 #72#
[gESRKS05TChKgm0NQDqtifSyhJZn+AyccF/TY3s7Hh4=]



Timestamp     2018-12-04 15:34:36.873, #81# [8c0EM0SkEZXhdoZc6ZGg9/cpIIpH3BHRs8kVdRO6Bj0=]
OpType        eIDAS Connector generates request to eIDAS Connector, #82# [zTkXH363Kr6iakmmQ+zXJS9XcSdYVSDhbZ1AHKC9Fac=]
NodeId        N/A, #83# [ECvzJGhlc+h0USJlbBXPhfzBA7CNYfkIHOKFqwY8S5U=]
Origin        N/A, #84# [mRJgVCy1XUYEbvW621OKbI/gRygVet8cyMRWVD7dw3o=]
Destination   N/A, #85# [9MKNwstKNFF7ED4H4vjYp/iR+FE+xv9h7Fl3qTmioJI=]
```

samlHash bZMkXeEoKHi2hFNTskYWefGo2+OxE0F7tpmA+i0du8whLUhu2ddwvBl0HvmADES/rhqhw+ib+aASp31Mt/T3Hg==, #86# [qxJ5rJt1yd97571KrUXzWsNeFuMbgqnZSje8ZZb9Fzo=]

originatingMsgId _5HzGYnqYAu8IVoEHAyxfSSQkGJb4HsUkuz6ZiggB2G.yhV4I_gZGOPIHyNb64D7, #87# [3nmrjKlHdQasxw4dox31MyzqzaTGHss3LR2wSTylmrk=]

msgId _VUQX-Z9q3VDtlyxOTCiNYbPvMAH3TBG1jO1LTiI3nyxy2rwIyx20p7MuOnPd49_ #88# [3O9fgwSX0DrOnJqqpCNRGzQ+hGP8yJZU1kTF7vb8yjE=]

Timestamp 2018-12-04 15:34:36.878, #90# [3X0+EJ5qz8zdftKxBUozifR9RbBlls+//tEMBpNqOfs=]

OpType eIDAS Connector sends request to eIDAS Service, #91# [H/tsR2Uvba5B3yrHmm64MgJdXNu7U6B/jFDvaUhxsw8=]

NodeId http://localhost:8080/EidasNode/ServiceMetadata, #92# [B0pEK8VN6NqiRE9Hr9UOKZ6Zs58t0rIHxzR7eW6kNzo=]

Origin N/A, #93# [rxUmHE53iDtdlhj0ZA9d9rI2LCOqoi/BQmUeddhr/5o=]

Destination http://localhost:8080/EidasNode/ColleagueRequest, #94# [fw4pK6UN/wbl2gkdZAJA1u/PxdbUWcuNg8/Bu704E5E=]

samlHash bZMkXeEoKHi2hFNTskYWefGo2+OxE0F7tpmA+i0du8whLUhu2ddwvBl0HvmADES/rhqhw+ib+aASp31Mt/T3Hg==, #95# [6an6LOmq3VwFc6sY27iWwMfpqGUqHsGqt6R+T13+PM4=]

originatingMsgId N/A, #96# [AyglrPpklRLLRC/yPMi/JIAkdxGska37KMkn64hvT6U=]

msgId _VUQX-Z9q3VDtlyxOTCiNYbPvMAH3TBG1jO1LTiI3nyxy2rwIyx20p7MuOnPd49_ #97# [dgxVlE1zjldDGt4J97wmyFzIkNRqKkxFpJuu8kDsr40=]

eIDAS Connector to eIDAS ProxyService

Timestamp 2018-12-04 15:34:38.56, #131# [0fndXJBZyioFiWubBc3WJE5EsueNXKLc2fM3NbPpSHM=]

OpType eIDAS Service receives request from eIDAS Connector, #132# [XptS4m41sAIYHIgzi3GzK/R4iQGsUnSek1V+ZJHp2RY=]

NodeId http://localhost:8080/EidasNode/ConnectorMetadata, #133# [cXqhIjnPs9FbBzbTFRflLLuJk0JIvx+tg5W4aDbtdVA=]

Origin http://localhost:8080/EidasNode/ServiceProvider, #134# [wGwFHYKldPmXK6QdGgYTJOVZFjnbgLH8zYJIhqdUsdA=]

Destination http://localhost:8080/EidasNode/ColleagueRequest, #135# [oW3ytrGeuxuVmP5bkPEr+fmTbRqwhDaaHHlJ2/Wm8Dg=]

samlHash bZMkXeEoKHi2hFNTskYWefGo2+OxE0F7tpmA+i0du8whLUhu2ddwvBl0HvmADES/rhqhw+ib+aASp31Mt/T3Hg==, #136# [SMKdOQGD281ZnUzB8x1P4jk9Og1JBClup6KU0gthLRU=]

originatingMsgId N/A, #137# [1eK8O1antm3kFyWaPzZYjRrPb94QY2X0rYL6ERgEkcs=]

msgId _VUQX-Z9q3VDtlyxOTCiNYbPvMAH3TBG1jO1LTiI3nyxy2rwIyx20p7MuOnPd49_ #138# [9ULHfQNJk5r3YxHdSAsEBEJjPlb3ErHVgj/u+xtcgF0=]

Timestamp 2018-12-04 15:34:48.41, #186# [Q7xNtxV87KkWqJUp7fS7xAwgi+KFEWBpRgTxnD6A+rw=]

```
OpType        eIDAS Specific Proxy Service generates request to IdP, #187# [25RJ0UYNNJFnE9IUfTSZ21TsEnCJLIRjI/98whSotDg=]
NodeId        N/A, #188# [Bs3KZFFNm4C8UsCmwBC1MS9DgpW4R/7XXxlC5OLtQnE=]
Origin        N/A, #189# [hNn2n8Ok+fJGvqCLuwrFTtlrDWSJGz1Pl2cqMtNhWcU=]
Destination   N/A, #190# [VM7gvfif8BXO+FNv+voFX4TnMwdOVtAyVHVIU1aeIhc=]
samlHash      bUVXXSrVoNU28d2GLMPRNl8t8vtrZyOvK4eb/Vbj1XbkH05edRk0BKahG91O5gMCgBcBfqSZVNwidwuxjy6dTw==, #191#
[kvLT0EaCbx9darlksYXAjtFZeZ9sQ0Tcgx6932ElKrA=]
originatingMsgId _VUQX-Z9q3VDtlyxOTCiNYbPvMAH3TBG1jO1LTiI3nyxy2rwIyx20p7MuOnPd49_, #192#
[IjrMTdYep68IFBNQO6AhJbgDdHXI04aRgZQhznSiPyI=]
msgId         _TNOcZhCXh-A5eU_Q5TPGFm.cl1QNAbBuiHGaeQNGJvTuO4a4lpEx8Kxjbxr.BtT #193#
[mY2icN87cjvKBcS7BtG1rDumRAkWp6BYTrSo9ltfQT0=]



Timestamp     2018-12-04 15:34:48.54, #204# [cjSijwhTJdSs+RWPTA6uoS4vCazAsmjRPzAbtyH/KcQ=]
OpType        eIDAS Specific Proxy Service sends request to IdP, #205# [gI3obw4vgm4A2tiWm4dz8LvOS2PUFI7+iZBrvZThENQ=]
NodeId        http://localhost:8080/IdP/metadata, #206# [z4UGcxHxY/809hDLskHn94XQX+lptNIitR+ZN0AvkCE=]
Origin        N/A, #207# [q3ONZKRleD/8WGZXtRJ6iwm2mCXz/KuUO5QCOJBD5JU=]
Destination   http://localhost:8080/IdP/AuthenticateCitizen, #208# [vTx8fCj40Plxk+4+1xF1imxLeBXxyizU4BQDGAV6xac=]
samlHash      bUVXXSrVoNU28d2GLMPRNl8t8vtrZyOvK4eb/Vbj1XbkH05edRk0BKahG91O5gMCgBcBfqSZVNwidwuxjy6dTw==, #209#
[oH4bGmpPSuu3ves72LlHycjqf+Zwis6Cd3t+vxV0Ob0=]
originatingMsgId N/A, #210# [9Z7zHMeZmPSm8NPLdUpmqyu3TfWYbHVvaHlmUNQSYjk=]
msgId         _TNOcZhCXh-A5eU_Q5TPGFm.cl1QNAbBuiHGaeQNGJvTuO4a4lpEx8Kxjbxr.BtT #211#
[EMLGeCQnxuP7C63kIiLevKja5PhtgxQxOzr3qrUHACE=]
```

eIDAS ProxyService to IdP

IdP Loggings

IdP to eIDAS ProxyService

```
Timestamp     2018-12-04 15:34:55.372, #231# [J92nZpDvoBWvTHSqpv68D3eslB22veFWw17k/9WcHfU=]
OpType        eIDAS Service receives response from IdP, #232# [hJuo/39Giqicr+NvRRLxpkQkZFSh8cPLFP8vcIEGmmY=]
NodeId        http://localhost:8080/IdP/metadata, #233# [R7J1ue2jHbSzlH3eWbH/Y24aWx5aDf9o3seOdoVzktM=]
Origin        http://localhost:8080/IdP/Login, #234# [VcGukfirtcbo1qwWXIuTEVdSTml1NmskyNFZrv66hKs=]
Destination   http://localhost:8080/EidasNode/IdpResponse, #235# [FUUVODpB4qSwYCpiBbb1G3AR4k+vxHbQcBAJYjjXMnk=]
```

```
inResponseTo  _TNOcZhCXh-A5eU_Q5TPGFm.cl1QNAbBuiHGaeQNGJvTuO4a4lpEx8Kxjbxr.BtT, #236#
[Mv3abYEFrOABld4MXIZs62/hqojh8jg5Hy+h8rhAVu0=]
statusCode    urn:oasis:names:tc:SAML:2.0:status:Success, #237# [pVCRZpE7fo/MbjH99njFH/VQV86q7VR4Ct0+Cr9LFPc=]
samlHash      izcLNPp0r5plFjf4Lzpazr0aZDVwaSMT7+vYYd4c24aA4TbIGpXypDZpgJUqj/eu4fh+WflelIwUEFFYNDetVw==, #238#
[BE5mUz9UcU5e398b7XiraqVaj1h+zRniuH3+njzvr6c=]
originatingMsgId N/A, #239# [xU/iazMQdUABR8h/0Nr/4VmcgANI1p6s7V8BIc9D14k=]
msgId         _LiLQNjneZERppaQXPsAL-bninh4jNBIe.2BjrozW.P3m5DOR1wrYeWPg6qVo9VW #240#
[c4s6qvG6BaUSNx7TZZueZVItytSciCrBt+9hXI6K9dk=]



Timestamp     2018-12-04 15:34:55.618, #242# [rvN8ECydb5pYcxgqmJ27PJmchyAe+QJNZQNS2ts7u7o=]
OpType        eIDAS Specific Service generates response to eIDAS Service, #243# [YG+6TT5/J3Krn9wAxVrDlCINYISU9E8kN8Yy68Q4wZA=]
NodeId        N/A, #244# [cvUZxMJd03dr0g1jyJzfCYxViJknLBuR3UH2GHjDoek=]
Origin        N/A, #245# [O9oZKn2V/jz0XxUiI/F468tuZ7/XnwO0ofXRuL5UHyQ=]
Destination   N/A, #246# [wAQwQyy2mml4M1FUmhzjEymoEgENwXz+JebC2RKPQsA=]
inResponseTo  _VUQX-Z9q3VDtlyxOTCiNYbPvMAH3TBG1jO1LTiI3nyxy2rwIyx20p7MuOnPd49_, #247#
[gRVsWeDMr0diWn+znYESqLsqKQq0XGyM8R0ygRPEoTI=]
statusCode    urn:oasis:names:tc:SAML:2.0:status:Success, #248# [jLB2OExZ6rGbCvJkyYvxsyeBpzMFpfd3QUCZpTescKY=]
samlHash      NbrFy4wyXUIuM72FOkU6v68zbVVuwG4Xrcj+CxtgSpDyvq8NHifXWm2Lo+scdnmur8ZkbQcKO5irS8sXRX5obA==, #249#
[H0HbTWHqsr5FAVVCXiCO7VXA5QmKo3k2z0HRKyv+Czo=]
originatingMsgId _LiLQNjneZERppaQXPsAL-bninh4jNBIe.2BjrozW.P3m5DOR1wrYeWPg6qVo9VW, #250#
[XOXNcY0BO1kJ1L07Nf7vhogXIJlS1gyRFMUWnn+4BFI=]
msgId         _ew3XUNM41yH4ZEwqT7nrAi7j7Lkw_BO7PWt5elbPhyD-bNNVyXyMHCkd1XcEZLu #251#
[Ug9WSLRziQOI1RPj3ZWWvbxdbetgGf13zp6j/lZuL98=]



Timestamp     2018-12-04 15:34:55.696, #256# [BAgPTdL7fTU4YHO64rcZL/ni94CBcqVpGe2unPRXx74=]
OpType        eIDAS Service generates response to eIDAS Service, #257# [4/1qHxJuvU9M6NyHCiEmzhwmfbzhmJP4pjXludn0n3w=]
NodeId        N/A, #258# [UK2BOxYLOVWCvWCfUtk4wCmkXaYLCQW8RLzRoCtoQp8=]
Origin        N/A, #259# [XNwYLUe+A4aTRNQW1J0agLcalpoM2AYDcvS8eGDpndI=]
Destination   N/A, #260# [0cclj5FH64U7vhkdX9xRVZjZw1JysKZtNWPYi/32OVw=]
inResponseTo  _VUQX-Z9q3VDtlyxOTCiNYbPvMAH3TBG1jO1LTiI3nyxy2rwIyx20p7MuOnPd49_, #261#
[3KPSxX6PYN4zB9QC/Z9IX7p/Q/NbsVQKFY+HtIsX8Ew=]
statusCode    urn:oasis:names:tc:SAML:2.0:status:Success, #262# [awEe270o6+Lzr+AQeWsIlg4P6Z6GtfvRRkwOJIhB8Lo=]
```

```
samlHash      1H4tRWSgpxL0GVqAGqsCPosth+ZiBMStf9WlaVqB9jnUJaSV/K1XCRKHyEaFvnFZ6P4m+BD10nR7DS+KtKu9PA==, #263#
[W+/0mAYOVNfNVW8B8Cf1s6eQsnZ4+moMNWbB0JJOXpA=]
originatingMsgId _ew3XUNM41yH4ZEwqT7nrAi7j7Lkw_BO7PWt5elbPhyD-bNNVyXyMHCkd1XcEZLu, #264#
[sKG8v0vvMu1ZwtOSNrFvSq1en/f+7wLNgSBxeVI3+T0=]
msgId         _jHKYJqiGA6fwGbrgxlOAd40eRzGsGHEDphWnw8or.pNlaq2VrPP.Ye2RUXI_KMf #265#
[pMmMQTtMwNS1/mZYvj746zBa4yMA8RMVmN5vVWdVJcI=]


Timestamp     2018-12-04 15:34:58.955, #285# [5iJ7ViYlhrAsvlGhpYutIlFXutk+yXBwvDmHA+mbdrs=]
OpType        eIDAS Service sends response to eIDAS Connector, #286# [Sh5xQJvUceZtWvcstL8Acnw0xEuVrLYbqPwoBkzeiZU=]
NodeId        http://localhost:8080/EidasNode/ConnectorMetadata, #287# [qyNZNJkl969cB0mx+DtWUFsJYHNd8KXeUy+jkbrDiYo=]
Origin        N/A, #288# [PfhhUJLorDdyagt6YJBX8Eb7yDLUU/nUBWtDxPW0PME=]
Destination   http://localhost:8080/EidasNode/ColleagueResponse, #289# [raT7+uzt3Hc5ps4dwCxcGYryJMOsAjcJ3K+uneVeKNs=]
inResponseTo  _VUQX-Z9q3VDtlyxOTCiNYbPvMAH3TBG1jO1LTiI3nyxy2rwIyx20p7MuOnPd49_, #290#
[26dD6iHtOVrnuYrSoqfmuE7pWbGQXoN3RSH32Wn5dF4=]
statusCode    urn:oasis:names:tc:SAML:2.0:status:Success, #291# [9xGd78FLOoD4b50iDNxh10xjjL/nrxYIbsAzhrcTuZU=]
samlHash      1H4tRWSgpxL0GVqAGqsCPosth+ZiBMStf9WlaVqB9jnUJaSV/K1XCRKHyEaFvnFZ6P4m+BD10nR7DS+KtKu9PA==, #292#
[J4Un7v+7fHfgwOou186N3WIBykRlUPmZEKzp7m78Pvs=]
originatingMsgId N/A, #293# [YJPp1dohDQts6mjYYeah479Ed3QHsw7nd/EpRlCv15A=]
msgId         _jHKYJqiGA6fwGbrgxlOAd40eRzGsGHEDphWnw8or.pNlaq2VrPP.Ye2RUXI_KMf #294#
[NNd5V1cfyCRjFiY+M+JtcqP2ZTRkaxKRTc/CCJs21ow=]


eIDAS ProxyService to eIDAS Connector


Timestamp     2018-12-04 15:34:59.342, #310# [u+CxVxj7kQ2sS+SRsaTMTp45bTvF/hrigE6GnwxIeAQ=]
OpType        eIDAS Connector receives response from eIDAS Service, #311# [NMyxLZnJOB4fB2q0Uf94rsV0gEwBumIVsxVHmPr1rIc=]
NodeId        http://localhost:8080/EidasNode/ServiceMetadata, #312# [n/Z0wb1TMjKBLZ5DHB9vCxxKHPEt5/nJ3tAnnEwu60o=]
Origin        http://localhost:8080/EidasNode/LogSaml, #313# [QJIItHDM/H1DF7WVEsENyfCrd9U57yeqSlPt2NDyiXQ=]
Destination   http://localhost:8080/EidasNode/ColleagueResponse, #314# [SdS69sMMPCBjDMzc4N076zP5EJpBSprFKmGjVJ+Betg=]
inResponseTo  _VUQX-Z9q3VDtlyxOTCiNYbPvMAH3TBG1jO1LTiI3nyxy2rwIyx20p7MuOnPd49_, #315#
[QtGoJCS359MjFVVyorX1ufpSJ/x7m0G/5d4NZfK2lvs=]
statusCode    urn:oasis:names:tc:SAML:2.0:status:Success, #316# [y0sQaoB/l9aajJkjMPJVSXjeX3vGkiEDiDcW2vXAprI=]
```

```
samlHash      1H4tRWSgpxL0GVqAGqsCPosth+ZiBMStf9WlaVqB9jnUJaSV/K1XCRKHyEaFvnFZ6P4m+BD10nR7DS+KtKu9PA==, #317#
[D/+0R2Tp79WorjGRu/PFZBHTpY/gp91k00XqGyiikL0=]
originatingMsgId N/A, #318# [EfthNicPJSbCHouL5eY1rjwHM9iLZRh4TCLIgRAxEwM=]
msgId         _jHKYJqiGA6fwGbrgxlOAd40eRzGsGHEDphWnw8or.pNlaq2VrPP.Ye2RUXI_KMf #319#
[R7gOTa6phf3sOpjafElhE2V6SHaCUrJ9pVYrdI8iRxg=]



Timestamp     2018-12-04 15:34:59.516, #324# [yh0ebNU7VewWDa9YMyqUGZH5FKPfcI5iIoqjrRH2nTs=]
OpType        eIDAS Specific Connector generates response to SP, #325# [gjNjPVwxUGLiS0rbf5xeqORSatowGLbPavsLfnq6Ft4=]
NodeId        N/A, #326# [UxeR6j5ie9gvpmIZ+8UrzMQhO/xJ7a/sawmI8+rvejU=]
Origin        N/A, #327# [9lRS+UVvqHTpm6/9IeiG4neayFUNnc1hrDJM1SLf288=]
Destination   N/A, #328# [r8SU37EGJ8z+vlwL4k48lxWCshpSxLYe0HaD6g0WLpU=]
inResponseTo  _d7GC643-eO9sCdegVOjEcXQNNk2ki6sNk2C7CNWsEi.SDNipIE2kwtU3d.k04Ht, #329#
[jJHVRe68rYDuh1YK0IYEO3hZQW3XMrO1wgsniDr3OKY=]
statusCode    urn:oasis:names:tc:SAML:2.0:status:Success, #330# [bvhxMUtF8eODAwUVniI+u0cjDzf/rcuAT29JMWco1C0=]
samlHash      z7D6shEWcPOfr01VFnJYMFTbxZn4C561niYdmPBVK9fDXqhNS0whjgDlbAc5+jp0Ke0Vt5fbbcSb1OOy1AEttA==, #331#
[pcbe15BGnYMYUFm+SbKkF2sfyqxkmm031YNDz3PKA7I=]
originatingMsgId _jHKYJqiGA6fwGbrgxlOAd40eRzGsGHEDphWnw8or.pNlaq2VrPP.Ye2RUXI_KMf, #332#
[2UraDcl/eCyp936KVu/XoI7ydClSnqhqbZ77JafYapk=]
msgId         _PDVLM55F-8b7kp7cYqXpY0qjeV83Xe8IWQuZRBh201Yoi6K3SZghr3kT9yrAvj3 #333#
[5u/kNcPv7NSkwL5wfAjSIRmoF24xtVuk8bijCV5S/Q8=]

Timestamp     2018-12-04 15:34:59.519, #335# [NFl/rqQC/CANUi9isZ9Q+wq+ZDbd8+YexQcVOPGfuCM=]
OpType        eIDAS Specific Connector sends response to SP, #336# [lmHvuqBaQNK2pwZBmJDLWvLEj8WR4NKB87v40aZgnrw=]
NodeId        http://localhost:8080/SP/metadata, #337# [81hI04/nl/O3KeoOTlf3DgLATrc3yH/DJ8bNhxPkCrU=]
Origin        N/A, #338# [mFNJ/FGoxm/IP4uUHuNG1f/6Y2yMfYqCBBpqph78VgQ=]
Destination   http://localhost:8080/SP/ReturnPage, #339# [W3ats0uCKtXAvOQLjX5T8r0TPcKx1Bo4KepnsrR4aBI=]
inResponseTo  _d7GC643-eO9sCdegVOjEcXQNNk2ki6sNk2C7CNWsEi.SDNipIE2kwtU3d.k04Ht, #340#
[oB0IGM4nVuMvLB9EKBAQMhto/8X+m1yuBGDYyIdvZ4E=]
statusCode    urn:oasis:names:tc:SAML:2.0:status:Success, #341# [vhlTDwGrnNbxei+GbCc4Vqyiwp+ChYGYZFcWK3mPQa4=]
samlHash      z7D6shEWcPOfr01VFnJYMFTbxZn4C561niYdmPBVK9fDXqhNS0whjgDlbAc5+jp0Ke0Vt5fbbcSb1OOy1AEttA==, #342#
[bYyJjeIArDgkx/6pquAY1IvQklgKmBJn/IBqUz5+YKU=]
originatingMsgId N/A, #343# [uoBoKx/TlBUmazMgs1/u9u5LRqTsUCqxhRG2qLetIck=]
msgId         _PDVLM55F-8b7kp7cYqXpY0qjeV83Xe8IWQuZRBh201Yoi6K3SZghr3kT9yrAvj3 #344#
[8l4bdx4IaOm7KjFfW3KFg1IO3o3n8X84qBw4EbKovaU=]
```

```
eIDAS Connector to SP

SP Loggings
```

4.2. How are the digests calculated in the logs

For integrity reasons, the logging makes use of hashes schemes for each log line and for the SAML messages exchanged. This section describes how the different digests in logging are calculated.

This may be useful for operators of eIDAS-Node to assess the integrity of subsequent logging entries, and for auditors to identify the message exchanged.

4.2.1. Calculate the digest of log lines

The digest algorithm used is SHA256.

For each log line the digest is calculated, using a hash-chaining scheme, and the corresponding digest values are appended to the log lines.

1. If this is not the first line in the log file, append the digest value of the previous line (in base64 format), together with '\n' character, to the new line for which the digest is to be calculated. Then, calculate the digest on this concatenated string.
2. If this is the first line in the log file, merely calculate the digest on this log line.
3. The digest of a line is base64 encoded.

Pseudocode

Excerpt of a pseudo code to calculate the digest for a log line.

```
newlineTxt = <text of a new log line>
if lineCount > 0 then
    newlineDgst = digest(prevlineDgstB64 + '\n' + newlineTxt)
else
    newlineDgst = digest(newlineTxt)
endif
newlineDgstB64 = base64(newlineDgst)
```

Example

Simplified example in python.

```
prevlineDgstB64 = b"1+OIWdSCX8F7eAlh/YEhR8kMUd5Urxu/Fo8WaDBOzzw="
newlineTxt = b"OpType        eIDAS Connector generates request to eIDAS Connector,
#97# "
h = hashlib.sha256()
h.update(prevlineDgstB64)
h.update(b'\n')
h.update(newlineTxt)
newlineDgst = h.digest()
newlineDgstB64 = base64.b64encode(newlineDgst)
print(newlineDgstB64)

--> ChmsHm9ZRn7VRsDhwSe+iOK5Ibf76VqAezEP6505hH0=
```

Thus, in the log file, the next line will follow the prev line, as shown below:

```
Timestamp     2018-11-12 10:44:14.454, #96# [1+OIWdSCX8F7eAlh/YEhR8kMUd5Urxu/Fo8WaDBOzzw=]
OpType        eIDAS Connector generates request to eIDAS Connector, #97#
[ChmsHm9ZRn7VRsDhwSe+iOK5Ibf76VqAezEP6505hH0=]
```

4.2.2. Calculate digest of the *samlHash* entry

The digest algorithm used is SHA512.

The digest is calculated for each conveyed SAML message. As the messages are base64 encoded when conveyed, before the digest they have to be base64 decoded.

Then, the digest is base64 encoded and put as value along with the corresponding samlHash log entry.

Note: No other processing (e.g., canonicalisation) is done on the message before the digest calculation.

Example

Simplified example in python.

```
msgXmlB64 = <SAML/XML message, base64 encoded>
msgXml = base64.b64decode(msgXmlB64)
h = hashlib.sha512()
h.update(msgXml)
msgDgst = h.digest()
msgDgstB64 = base64.b64encode(msgDgst)
```

5. Logging configuration settings

You should ensure that other, guidelines, and procedures that have some relationship to logging incorporate and support these log management requirements and recommendations, and also comply with functional and operational requirements. The following table provides examples of the types of logging configuration settings to be specified in policies which are MS specific.

Table 2: Logging configuration settings

Category	Low Impact Systems	Moderate Impact Systems	High Impact Systems
How long to retain log data	1 to 2 weeks	1 to 3 months	3 to 12 months
How often to rotate logs	Optional (if performed, at least every week or every 25 MB)	Every 6 to 24 hours, or every 2 to 5 MB	Every 15 to 60 minutes, or every 0.5 to 1.0 MB
If the organisation requires the system to transfer log data to the log management infrastructure, how frequently that should be done	Every 3 to 24 hours	Every 15 to 60 minutes	At least every 5 minutes
How often log data needs to be analysed locally (through automated or manual means)	Every 1 to 7 days	Every 12 to 24 hours	At least 6 times a day
Whether log file integrity checking needs to be performed for rotated logs	Optional	Yes	Yes
Whether rotated logs need to be encrypted	Optional	Optional	Yes
Whether log data transfers to the log management infrastructure need to be encrypted or performed on a separate logging network	Optional	Yes is feasible	Yes

6. Log files

Each eIDAS Node is configured to generate four different locally stored log files for tracing activities:

- **The Application System log:** (`eIDASNodeSystem`) contains all the events occurring at the application level (e.g. server start/stop, change of configuration).
- **The Application Security log**: (`eIDASNodeSecurity`) contains all the security related events [8].
- **The Message Exchange log:** (`eIDASNodeSAMLExchange`) contains the details about exchanged messages.
- **The Application Detailed log**: (`eIDASNodeDetail`) contains the more detailed information at destination to the technical team for forensics, investigation and debugging purposes.

The locations of the audit files are by default configured to use a Java system properties variable called `LOG_HOME`.

A value can be assigned to this variable by using: `-DLOG_HOME="<myDirectoryName>"` at server start-up.

If modification of the environment variable is not possible, the value of this variable could also be assigned by adding the following line in the `logback.xml` file
`<property name="LOG_HOME" value ="<myDirectoryName>" />`

The following table shows a matrix of which events are logged, in which log file and which response would be triggered.

Table 3: Event log matrix

		Record in log file						Response type			
Event	**System**	**Detailed**	**Security**	**Message Exchg**	**Console**	**Appl. server log**	**Threshold level (see Table 1)**	**Immediate**	**Routine**	**Forensics - historic**	**Comment**
SAML request processing [EVT-1]		√		√	√		Info		√		Normal processing of the SAML messages
SAML Response processing [EVT-2]		√		√	√		Info		√		Normal processing of the SAML messages
OpenSAML Error [EVT-3]		√		√	√		Error	√			Error occurs in the OpenSAML product library
Configuration change [EVT-4]	√				√				√	√	Configuration change via management console
Configuration reload [EVT-5]	√				√				√	√	
Application startup [EVT-6]	√					√			√	√	Start of application filters
Application shutdown [EVT-4]	√					√			√	√	Start of application filters
Console alerts [EVT-5]	√	√			√			√		√	

		Record in log file						Response type			
Event	**System**	**Detailed**	**Security**	**Message Exchg**	**Console**	**Appl. server log**	**Threshold level (see Table 1)**	**Immediate**	**Routine**	**Forensics - historic**	**Comment**
Exceptions Application errors and system events e.g. syntax and runtime errors, connectivity problems, performance issues, file system errors [EVT-6]		√			√			√			
Unauthorised access attempt to management console [EVT-7]		√	√		√			√			
Input validation failures [EVT-8]		√	√		√				√		Could hide an attack
Output validation failures [EVT-9]		√	√		√				√		Could hide an attack
Session management failures e.g. cookie session identification value modification [EVT-10]		√	√		√			√			Could hide an attack
Content security violation [EVT-11]		√	√		√			√			Could hide an attack

		Record in log file						Response type			
Event	**System**	**Detailed**	**Security**	**Message Exchg**	**Console**	**Appl. server log**	**Threshold level (see Table 1)**	**Immediate**	**Routine**	**Forensics - historic**	**Comment**
Sequencing failure (like SAML Response without being preceded by SAML Request) [EVT-12]		√	√		√			√			Could hide an attack
Excessive use (like SAML request limitation exceeded) [EVT-13]		√	√		√			√			Could hide an attack
Certificate not valid [EVT-13]		√	√		√			√			Could hide an attack
Signature not valid [EVT-14]		√	√		√			√			Could hide an attack

6.1. Event detailed error codes and associated actions

For full information on eIDAS-Node error codes and associated actions, refer to the *eIDAS-Node Error Codes* document [6].

7. Operational considerations

Ref: [1], [3], [7], [8], [9], [11]

- Audit facilities should be designed with their specific use(s) in mind, not simply adapted from existing system logs.
- Member States should create and maintain an audit log management structure, including policy, procedures, rules and tools.
- Audit log files should be periodically saved and moved to another storage space. The periodicity may be influenced by the classification of the data, but should be defined either according to a fixed size (e.g. every time the log reaches 1Mb) and / or according to a fixed period of time (e.g. per day, per week).
- Attacks on systems should be prevented by the use of technical preventive controls. These controls should be preferred above procedural preventive controls if possible. Both measures should be used in conjunction with examination of the audit data.
- Member States should consider the use of a Security Event management system due to the large volume of security events.
- Member States should not solely rely on unauthorised events or breaches of policy, but should additionally consider proactive monitoring.
- Member States should consider the use of Security Information and Event Management (SIEM) technology to help the automation and collection of auditable events.
- Audit records should contain a time stamp. Ensure that each system's clock is synchronised to a common time source so that its timestamp will match those generated by other systems.
- The privacy of personal data must be protected [10].
- Measures should be implemented in order that deleting or modifying logs of own activities should be detected and alerted immediately.
- Administrators should not be able to erase or de-activate logs of activities.
- Audit events should be protected from modification by using digital signatures to sign audit records.
- Audit events should be archived to 'write-once 'media to protect the archive from modification or deletion.
- Where audit records are subject to cryptographic authentication the input data to the authentication computation should include an accurate timestamp.
- Archived audit log files should be protected by appropriate logical and physical security mechanisms.
- The capture of customer sensitive data in audit logs should be avoided. If necessary sensitive data should be anonymised, tokenised or encrypted to avoid data breaches.
- Where audit data are encrypted, the appropriate decryption software and keys must be available and properly maintained, under appropriate access control, for the whole of the lifetime of the encrypted data. Proper maintenance should include periodic testing and adequate back-up to cater for loss of stored encryption keys.
- Audit records should be created in a simple standard format.

- Where audit data are compressed, the appropriate decompression software must be available and properly maintained for the whole of the lifetime of the compressed data. Proper maintenance should include periodic testing.
- The audit trails [11] [12] must be stored in a long-term support and a tamper evident format, such that illicit addition, modification or deletion of any audit trail can be detected.

7.1. Retention period

- MS should ensure that audit retention is part of the organisation's overall retention policy.
- Where the execution of sensitive security-related actions cannot be made subject to dual control, then that execution should be monitored in a timely fashion [9].
- MS should determine which data is classified as audit data and should protect and preserve this data appropriately.
- Personal notes should be kept by incident investigators throughout the whole course of an investigation and those notes should themselves be protected and preserved in the same manner as audit data.
- Audit records used during an investigation must be preserved at least until the end of the investigation and for any subsequent prosecution irrespective of their normal retention period.
- Follow the organisation's incident response policy to investigate an audit log incident.
- System configuration information should be modified, if necessary, to prevent an event from overwhelming the system.

8. References

[1] NIST: An Introduction to Computer Security: The NIST Handbook, NIST Special Publication 800-12, December 1997, http://csrc.nist.gov/publications/nistpubs/800-12/handbook.pdf

[2] eIDASInteropArch: eIDAS - Interoperability Architecture v1.0, DIGIT, 6 November 2015, https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eIDAS+Profile?preview=/46992719/47190130/eidas_interoperability_architecture_v1.00.pdf

[3] SANS Consensus Policy Resource Community - Information Logging Standard, http://www.sans.org/security-resources/policies/server-security

[4] COBIT (Control Objectives for Information and related Technology) from Information Systems Audit and Control Association (http://www.isaca.org/cobit.htm)

[5] ISO/IEC 27002 - Information technology -- Security techniques -- Code of practice for information security management, section 10.10, 2005 (www.iso.org)

[6] eIDAS-Node Error Codes, v1.4.4, DIGIT, 14th December 2018.

[7] BSI PD008: Legal Admissibility and Evidential Weight of Information Stored Electronically, British Standards Institution, 1999

[8] K. Kent, M. Souppaya. Guide to Computer Security Log Management. Recommendations of the National Institute of Standards and Technology, NIST Special Publication 800-92, September 2006

[9] Common Criteria: Common Criteria for Information Technology Security Evaluation, Version 3.1, revision 4, September.2012 Part 2: Security Functional Components, http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R4.pdf

[10] ENISA: Privacy Features of European eID Card Specification, Version 1.0.1, January 2009, http://www.enisa.europa.eu/doc/pdf/deliverables/enisa_privacy_features_eID.pdf

[11] AUDIT Trails - NIST publication http://csrc.nist.gov/publications/nistbul/itl97-03.txt

[12] EC council - The use of audit trails in security systems http://www.europeanpaymentscouncil.eu/index.cfm/knowledge-bank/epc-documents/the-use-of-audit-trails-in-security-systems-guidelines-for-european-banks/

Appendix A. eIDAS-Node Error Codes and Error Messages

The following table provides a list of eIDAS-Node errors and related error messages.

Table 4: Error Codes and Error Messages

Key	Description
sProviderAction.invalidSPQAA.code	Invalid SAML Parameter SP QAA Level error code on the ServiceProvider interface.
sProviderAction.invalidSPQAA.message	Invalid SAML Parameter SP QAA Level error message on the ServiceProvider interface.
sProviderAction.invalidQaaSPid.code	Invalid SP QAA Level error code on the ServiceProvider interface.
sProviderAction.invalidQaaSPid.message	Invalid SP QAA Level error message on the ServiceProvider interface.
sProviderAction.invalidSpId.code	Invalid SP ID error code on the ServiceProvider interface.
sProviderAction.invalidSpId.message	Invalid SP ID error message on the ServiceProvider interface.
domain.ServiceProviderAction.code	Invalid SP domain error code on the ServiceProvider interface.
domain.ServiceProviderAction.message	Invalid SP domain error message on the ServiceProvider interface.
sProviderAction.invalidSPDomain.code	Invalid SP Domain error message on the ServiceProvider interface.
sProviderAction.invalidSPDomain.message	Invalid SP Domain error message on the ServiceProvider interface.
sProviderAction.invalidCountry.code	Invalid Country selected by the Citizen error message on the ServiceProvider interface.
sProviderAction.invalidCountry.message	Invalid SP Domain error message on the ServiceProvider interface.
sProviderAction.spNotAllowed.code	SP not allowed error code on the ServiceProvider interface.
sProviderAction.spNotAllowed.message	SP not allowed error message on the ServiceProvider interface.
sProviderAction.invalidSaml.code	Invalid SP Saml request error code on the ServiceProvider interface.
sProviderAction.invalidSaml.message	Invalid SP Saml request error message on the ServiceProvider interface.
sProviderAction.invalidSPProviderName.code	Invalid SP Provider Name error code on the ServiceProvider interface.
sProviderAction. invalidSPProviderName.message	Invalid SP Provider Name error message on the ServiceProvider interface.
sProviderAction.invalidSPRedirect.code	Invalid SP Redirect URL parameter error code on the ServiceProvider interface.
sProviderAction.invalidSPRedirect.message	Invalid SP Redirect URL parameter error message on the ServiceProvider interface.
sProviderAction.invalidRelayState.code	Invalid SP Relay State parameter error code on the ServiceProvider interface.
sProviderAction.invalidRelayState.message	Invalid SP Relay State parameter error message on the ServiceProvider interface.

Key	Description
sProviderAction.invalidAttr.code	Invalid SP personal attribute list error code on the ServiceProvider interface.
sProviderAction.invalidAttr.message	Invalid SP personal attribute list error message on the ServiceProvider interface.
requests.ServiceProviderAction.code	"Maximum number of requests from SP allowed" error code on the ServiceProvider interface.
requests.ServiceProviderAction.message	"Maximum number of requests from SP allowed" error message on the ServiceProvider interface.
sProviderAction.invalidSPAlias.code	Invalid SP Alias error code on the ServiceProvider interface.
sProviderAction.invalidSPAlias.message	Invalid SP alias error message on the ServiceProvider interface.
sProviderAction.errorCreatingSAML.code	Generating SAML Token error code on the ServiceProvider interface.
sProviderAction.errorCreatingSAML.message	Generating SAML Token error message on the ServiceProvider interface.
internalError.code	Generic error code.
internalError.message	Generic error message.
attrList.code	Invalid attribute list error code on Specific code.
attrList.message	Invalid attribute list error message on Specific code.
missing.sessionId.code	Missing session id error code.
missing.sessionId.message	Missing session id error code.
invalid.sessionId.code	Invalid session id error code.
invalid.sessionId.message	Invalid session id error code.
invalid.session.code	Invalid session error code.
invalid.session.message	Invalid session error code.
callback.code	Invalid eIDAS-Node Proxy Service callback URL error code on CitizenConsent interface.
callback.message	Invalid eIDAS-Node Proxy Service callback URL error message on CitizenConsent interface.
idp.url.code	Invalid IdP redirect URL error code.
idp.url.message	Invalid IdP redirect URL error message.
IdPSAMLResponse.code	Invalid IdP's SAML Response error code.
IdPSAMLResponse.message	Invalid IdP's SAML Response error message.
serviceSAMLResponse.code	Invalid Proxy Service's SAML Response error code.
serviceSAMLResponse.message	Invalid Proxy Service's SAML Response error message.

Key	Description
authenticationFailed.code	Authentication Failed error code on IdPResponse Interface.
authenticationFailed.message	Authentication Failed error message on IdPResponse Interface.
username.code	Authentication Failed error code on specific code.
username.message	Authentication Failed error message on specific code.
invalidAttributeList.code	Invalid Attribute List error code on Specific Code or IdP Response.
invalidAttributeList.message	Invalid Attribute List error message on Specific Code or IdP Response.
invalidAttributeValue.code	Invalid Attribute Value error code on Specific Code.
invalidAttributeValue.message	Invalid Attribute Value error message on Specific Code.
attVerification.mandatory.code	No consent given error code on the CitizenConsent interface.
attVerification.mandatory.message	No consent given to mandatory error message on the CitizenConsent interface.
attValue.mandatory.code	Missing attribute value error code.
attValue.mandatory.message	Missing attribute value error message.
hash.error.code	Hash generation error code.
hash.error.message	Hash generation error message.
qaaLevel.code	Invalid QAALevel error code on Specific Code or IdP Response.
qaaLevel.message	Invalid QAALevel error message on Specific Code or IdP Response.
colleagueRequest.invalidSAML.code	Invalid SAML Token error code on the ColleagueRequest interface.
colleagueRequest.invalidSAML.message	Invalid SAML Token error message on the ColleagueRequest interface.
colleagueRequest.invalidCountry.code	Invalid Citizen Country error code on the ColleagueRequest interface.
colleagueRequest.invalidCountry.message	Invalid Citizen Country error code on the ColleagueRequest interface.
colleagueRequest.errorCreatingSAML.code	Generating SAML Response error code on the ColleagueRequest interface.
colleagueRequest.errorCreatingSAML.message	Generating SAML Response error code on the ColleagueRequest interface.
colleagueRequest.invalidQaa.code	Invalid SP QAA Level error code on the ColleagueRequest interface.
colleagueRequest.invalidQaa.message	Invalid SP QAA Level error message on the ColleagueRequest interface.
colleagueRequest.attrNull.code	Invalid personal attribute list error code on the ColleagueRequest interface.
colleagueRequest.attrNull.message	Invalid personal attribute list error message on the ColleagueRequest interface.

Key	Description
colleagueRequest.invalidRedirect.code	Invalid SP return URL error code on the ColleagueRequest interface.
ColleagueRequest.invalidRedirect.message	Invalid SP return URL error message on the ColleagueRequest interface.
colleagueRequest.invalidDestUrl.code	Invalid Citizen Consent URL error code on the ColleagueRequest interface.
colleagueRequest.invalidDestUrl.message	Invalid Citizen Consent URL error code on the ColleagueRequest interface.
serviceRedirectUrl.code	Invalid Proxy Service Redirect configuration URL error code on the ServiceProvider interface.
serviceRedirectUrl.message	Invalid Proxy Service Redirect configuration URL error message on the ServiceProvider interface
service.attrNull.code	Invalid personal attribute list on eIDAS-Node Proxy Service error code.
service.attrNull.message	Invalid personal attribute list on eIDAS-Node Proxy Service error message.
citizenNoConsent.mandatory.code	No consent given error code on the ConsentType interface.
citizenNoConsent.mandatory.message	No consent given to mandatory error message on the ConsentType interface.
colleagueResponse.invalidSAML.code	Invalid SAML Token error code on the ColleagueResponse interface.
colleagueResponse.invalidSAML.message	Invalid SAML Token error message on the ColleagueResponse interface.
citizenResponse.mandatory.code	No consent given error code on the ConsentType interface.
citizenResponse.mandatory.message	No consent given to mandatory error message on the ConsentType interface.
auRequestIdError.code	The SAML Response Id is either null or not valid error code.
auRequestIdError.message	The SAML Response Id is either null or not valid error message.
audienceRestrictionError.code	SAML Conditions are not compliant error code.
audienceRestrictionError.message	SAML Conditions are not compliant error message.
connectorSAMLResponse.code	Generating eIDAS-Node Connector SAML Response error code on the ColleagueResponse interface.
connectorSAMLResponse.message	Generating eIDAS-Node Connector SAML Response error message on the ColleagueResponse interface.