# INTRODUCTION

## INFORMATION FROM THE TECHNICAL EVALUATION REPORT

|  |  |
| --- | --- |
| **Reference component** | IDSA Reference Testbed |
| **Version** | 1.0 |
| **Author or authors** | Fraunhofer AISEC, Fraunhofer ISST and Fraunhofer IAIS |
| **Approved by** |  |
| **Date** |  |
| **File code** |  |
| **Type of evaluation** |  |
| **Taxonomy of the product** |  |

## DEVELOPER AND TOE INFORMATION

|  |  |
| --- | --- |
| **Developer Data (Name and Address)** |  |
| **Developer Contact Details (Name and e-mail)** |  |
| **Name of the TOE** |  |
| **Version of the TOE** |  |

# DECRIPTION OF THE TOE

The reference testbed is currently comprised of the three base components required to make a minimum viable interoperability infrastructure. These components are a Connector, a Dynamic Attribute Provisioning Service (DAPS) and a Broker. The Connector is the central technical component of the International Data Spaces and it is capable of fulfilling both data provider and data consumer roles.

## FUNCTIONAL DESCRIPTION OF THE TOE

## INVENTORY OF SECURITY FUNCTIONS IDENTIFIED IN THE SECURITY STATEMENT

Integrity checks, encryption, PKI… etc

# EXECUTION ENVIRONMENT

## DESCRIPTION OF THE EXECUTION ENVIRONMENT

The reference testbed is being tested in Linux, Ubuntu 20.04. To launch the different components included we will need different kinds of software.

The Dataspace Connector makes use of Git, Maven (tested on 3.6.3) and requires at least Java 11.

The Metadata Broker makes use of Git, Docker, Docker compose and OpenSSL.

The Omejdn DAPS makes use of Git, Docker and Ruby.

To be able to successfully install the testbed the user requires a system with:

Linux OS: Ubuntu 20.04  
Java 11 (at least)  
Maven  
Git  
Docker  
Docker Compose  
Ruby  
OpenSSL

## HIPÓTESIS SOBRE EL ENTORNO DE EJECUCIÓN

# EXECUTIVE SUMMARY OF THE EVALUATION

# EVALUATION VEREDICT

# COMPONENT INSTALLATION

The Testbed installation will be broken down into two steps:  
1. The components are individually installed and tested  
2. The components are configured to interact with each in a closed environment

# CONFORMITY ANALYSIS

## ANALYSIS OF THE SECURITY STATEMENT

## ANALYSIS OF DOCUMENTATION

## PROVEN FUNCTIONALITIES

|  |  |  |  |
| --- | --- | --- | --- |
| **Test code** | | (Ej.- TEST\_0xx) | |
| **Proven functionality:** Identificación de ataque | | Evaluator: | |
| Objetive of the test: Un detector de intrusiones debe de ser capaz de detectar ataques conocidos y debería disponer de las firmas/reglas necesarias para ello. | |
| **Test scenario:** máquina recién instalada | | | |
| **Procedure** | **Expected results** | | **Results** |
| Se realizará un ataque al servicio Apache 1.3.20 aprovechando la vulnerabilidad conocida. Para ello se habilitará una  máquina externa que lanzará el ataque de | El IDS debe de detectar cuando el ataque se ha producido con marcas de tiempo similares a las de la máquina atacante. | | Se puede observar en la información proporcionada por el IDS que se han detectado todos los ataques  lanzados, pero, en algunos casos, con retardos |
| **Conclusion and verdict** | | | |
|  | | | |

# VULNERABILITY SCANNING

# REFERENCES

# ACRONYMS