Cloud Orchestration at the Level of Application

Project Acronym: **COLA**

Project Number: **731574**

Program: **Information and Communication Technologies**
Advanced Computing and Cloud Computing

Topic: **ICT-06-2016 Cloud Computing**

Call Identifier: **H2020-ICT-2016-1**

Funding Scheme: **Innovation Action**

Start date of project: 01/01/2017 Duration: 30 months

Deliverable:

D7.6 MiCADO Security Evaluation Report

Due date of deliverable: 30/09/2019 Actual submission date: 22/09/2019

WPL: Nicolae Paladi

Dissemination Level: PU

Version: 1.1

Status and Change History

Table 1 Status Change History

Status:	**Name:**	**Date:**	**Signature:**
Draft:	A. Michalas and N. Paladi	10/06/2019	A.Michalas/N. Paladi
Reviewed:	M. Kendall	06/09/2019	M. Kendall
Approved:	T. Kiss	22/09/2019	T. Kiss

Table 2 Document Change History

Version	**Date**	**Pages**	**Author**	**Modification**
V0.1	02/04/2019	38	Amjad, Hai-Van	First draft
V0.2	05/04/2019	37	Antonios	Revised first draft
V0.21	10/04/2019	37	Nicolae	Review first draft
V0.3	17/04/2019	37	Amjad, Hai-Van	Revised draft after Nicolae's review
V0.4	17/06/2019	59	Denes Fodor, Balint Kovacs	Added penetration test report
V0.5	24/06/2019	59	Amjad	Rectified some mistakes identified during COLA project meeting.
V0.6	09/07/2019	62	Amjad	Add "Evaluation based on Main Security requirement" section.
V0.61	15/07/2019	65	Nicolae	Add introduction and conclusion, review document
V0.7	31/07/2019	65	Amjad, Hai-Van	Added "Security performance evaluation" section, performed integration of various sections introduced in other versions of document that caused versions conflict.
V0.8	16/08/2019	66	Nicolae	Review complete draft, submit for internal review
V0.9	09/09/2019	66	Amjad, Hai-Van	Resolve comments from the internal reviewer; fix formatting mistakes
V1.0	11/09/2019	65	Nicolae	Review latest changes, final checks.
V1.1	12/09/2019	66	Nicolae	Fix index for tables and figures

List of Figures and Tables

Tables

Figures

Table of contents

Glossary

Table 3 Glossary

CCSR	Cloud Compute Security Requirements
CNSR	Cloud Network Security Requirements
CSSR	Cloud Storage Security Requirements
COLA	Cloud Orchestration at the Level of Application
MiCADO	Microservice-based Cloud Application-level Dynamic Orchestrator
IIVR	Image Integrity Verifier
SPM	Security Policy Manager
CSP	Cloud Service Provider
PII	Personally, Identifiable Information
WN	Worker node
MN	Master node
TLS	Transport Layer Security
IPsec	Internet Protocol Security

1 Introduction

Throughout Project COLA, activities in work package (WP) 7 focused on enhancing the security of the MiCADO framework. This work contributed to advancing the state of the art in the area of cloud orchestration security and resulted in a set of security enablers designed, developed and integrated in the MiCADO framework. As a result, the MiCADO framework comprises additional capabilities to protect the integrity and authenticity of cloud federated cloud deployments.

Security work in WP7 was done in several distinct stages and each stage contributed reusable building blocks for the security architecture of MiCADO. The stages were:

- COLA Security Requirements (described in Deliverable D7.1)
- MiCADO security architecture specification (described in Deliverable D7.2)
- MiCADO application security classification specification (described in Deliverable D7.3)
- MiCADO security policy formats specification (described in Deliverable D7.4)
- MiCADO security modules reference implementation (delivered and described in Deliverable D7.5)

The current deliverable is the last installment in the series of the deliverable of WP7. The **goal** of this document is to describe the security evaluation of the security enablers developed and delivered within WP7 of project COLA. The evaluation of the enablers is described in terms of:

- Security Requirements collected from the COLA use case partners;
- Evaluation based on NIST 800-53 compliance control reference
- Evaluation of the security configuration guidelines
- Penetration testing of the enablers developed within WP7.

The evaluation provides a comprehensive overview of the state of the security enablers developed within WP7 and implicitly of the security of MiCADO. However, it is worth noting that the security landscape is continuously evolving. Therefore, new threats will emerge – both in aspects addressed by the security enablers and in aspects that were not prioritized by the use case partners and hence not addressed by any security enablers. This deliverable offers a snapshot of the security of MiCADO at the current stage.

2 Security Features Evaluation

This section discusses the evaluation of MiCADO security features. The evaluation is based on MiCADO v0.7.2 – the latest release at the time of writing this deliverable. The evaluation consists of the following:

- Assessments of the MiCADO framework based on the security requirements collected from use-cases and the main security requirements identified in the previous deliverable D7.1 [3].
- Assessment of the MiCADO framework as per the standards defined by NIST for Federal Information Systems and Organization [8].

In both cases, we highlight and discuss the relevant security features of MiCADO framework along with their description and implementation status. In addition to that, the relevant guidelines, prefixed with GL such as GL1, GL2 (see Section 2.4), are specified against each security feature, when applicable. The implementation status can be one of the following types:

1. **Supported:** This reflects that MiCADO framework support features that comply with the requirement.
2. **Partial:** This reflects that either MiCADO framework support features partially comply with the requirement; or MiCADO supports features that fully comply with the requirement, however these features are not enabled in the current release.
3. **Not available:** The feature is currently not in place. However, it is considered as an addition in the future versions of MiCADO.
4. **Not applicable:** The underlying feature is not under the direct control of MiCADO framework and the application owners should take care to comply with the any necessary requirements.
5. **None:** The underlying feature is not supported by MiCADO. The key reasons include: (1) The underlying requirement is mainly related to information systems in general. However, MiCADO is not considered as such a framework, or (2) MiCADO does not support relevant features as they are not required.

2.1 Evaluation based on Security Requirements from use cases

This section discusses and evaluates the applicability of MiCADO framework based on its success in addressing and fulfilling the security requirements/expectations of its target users. In this regard, various security requirements from the target application domains were identified previously from different end-user target organizations. The details of these organizations and their security specific requirements can be found in Deliverable D7.1 [3]. The following subsections analyse the behaviour of the MiCADO framework in accordance to the security requirements provided by the target end-user organizations including *Inycom*, *SAKER*, *Outlandish*, and *CloudSME*.

2.1.1 Inycom

Inycom provides high quality services and solutions with added value in IT and Communications, Energy, Laboratory Equipment, Electronics and Medical Equipment. The security requirements obtained from *Inycom are the following*[1]*:*

[1] The implementation of MiCADO framework is analysed and evaluated to reflect on the achievement of these requirements.

1. ***End users need a user/password to access the web interface:*** MiCADO is a generic framework that supports dynamic application level orchestration of cloud applications. Hence, it manages applications, their deployment and the underlying infrastructure; and has no control and responsibility to handle internal details of the deployed applications. MiCADO, however, facilitates secure management of those users who are responsible for the deployment and management of applications through various security components including Credential Manager and L7 firewall (i.e. Zorp) (refer to D7.5 [7] for full details). However, these users are not end users who will use the Inycom's applications (or any application in general). To summarize, the management of end users and their authentication to the application is directly handled by the applications themselves and is therefore out the scope of the MiCADO framework. On the other hand, MiCADO facilitates secure access for the users of the system, using a traditional password-based authentication mechanism.
2. ***Option to transfer encrypted data in the case of personally identifiable information (PII) – such as citizens' data:*** MiCADO framework facilitates data protection in different aspects. The following description summarizes the protection of various types of data in the context of MiCADO. However, it is worth mentioning that this requirement is *not* mandatory as per the description of the use-case (refer to section 9.1.1 of D7.1 [3]):
   A) **Data needed for deploying the application, e.g. private docker registry account information.** This data is fetched from a MiCADO user who deploys her application in the framework to the Master Node. This data is then sent from the Master Node to the Worker Nodes that hosts the application containers. During transit, the data is protected through TLS communication between MiCADO users and the Master Node, and IPSec communication between the Master Node and Worker Nodes.
   B) **Application data communicated from/to end users.** To protect this type of data, TLS must be configured for nodes hosting the frontend services to which the end users connect. This is an application specific requirement and therefore it should be handled by the actual applications. However, MiCADO can be easily extended to facilitate applications to setup TLS support, if required, using the Application Description Template (ADT) based configuration. This can be done via the NetworkSecurityPolicy (see Section 3.2 and Deliverable 7.4 [6] for further details), and application secret mechanism for holding the key and certificate. Specifically, NetworkSecurityPolicy allows specifying a network protocol enforcement, an application-level firewall as well to provide TLS control at worker nodes. Although MiCADO currently allows the definition of such a policy and the application secret mechanism is available, the automated configuration of TLS in worker nodes was not implemented yet. This requirement is *partially* supported.

   Apart from the above-mentioned scenarios, the explicit in/out transfer of application data in any case is not in the scope of MiCADO. If transfer is required, then the application user itself should fulfil the requirement.
3. ***Databases will offer restricted access (i.e. will only be accessible from a limited set of IPs):*** The use of databases in MiCADO are of the following two kinds: *(1)* The databases can be deployed as containers in Worker Nodes, or *(2)* Databases can be deployed outside MiCADO's controlled infrastructure. In the former case, worker nodes are configurable using L7 filtering to allow access to databases only for a certain set of IPs. More specifically, the application owner can define the whitelisted IPs as an attribute of the policy *L7Proxy* in ADT template (refer to Section 3.2,

Deliverable 7.4 [6] for further details). By doing this, databases will be only accessible from a trusted list of IPs. Thus, this requirement will be fully satisfied when databases are deployed and managed by the MiCADO framework. However, this feature is not active yet and therefore, the underlying requirement is considered as *partially* supported. In the case where databases are deployed outside of the MiCADO framework, protecting and managing access is not the responsibility of the MiCADO framework. Instead, it should be protected and managed by the hosting infrastructure.

4. ***Data will be stored in EU or associated countries with data protection regulations at least as restrictive as the EU one:*** MiCADO empowers users to describe application(s), infrastructure, and policies using an easy-to-use TOSCA-based ADT during the deployment of an application into the framework. Among the provided information, the user has to provide the infrastructure related deployment id, which represents the details of the infrastructure. Hence, the users have the ability to choose the infrastructure by themselves that will be used by the application and for data storage. Once this information is provided, MiCADO complies with these configurations for the entire execution time.

Based on the above analysis, the assessment results against Inycom's use-case requirements are summarised in Table 5.

Table 4 MiCADO assessment in accordance to Inycom's use-case requirements

Inycom use-case requirement	Status	Guideline
End users need a user/password to access the web interface	Supported	GL5
Option to transfer encrypted data in the **case** of **personally identifiable information (PII) – such as citizens' data** [Not mandatory]	Partial	
Databases will only be accessible from a restricted list of IPs	Partial	
Data will only be stored in EU or associated countries with data protection regulations at least as restrictive as the EU one	Not Applicable	

2.1.2 SAKER

SAKER solutions ltd is a provider of simulation-based tools, training, support and consultancy in the UK. The security requirements obtained by SAKER *are the following*[2]*:*

1. ***The system must be able to run on a private network or private cloud: The*** MiCADO framework supports the deployment and orchestration of applications on private cloud environments. It has been tested and utilized extensively for OpenNebula and OpenStack based private cloud environments. TOSCA-based ADT templates for using the MiCADO framework over OpenNebula and OpenStack environments have been made publicly available with the open source release of

[2] The implementation of MiCADO framework is analysed and evaluated to reflect on the achievement of these requirements.

MiCADO. This can be obtained from the MiCADO-Scale public repository over Github [9]. Hence, the MiCADO framework *fully* satisfies this requirement.

2. ***File encryption may be seen as a requirement depending on the client and application:*** Data protection can be considered at two different stages, i.e. in transit and at rest. Data in transit between users and the master node is protected by building secure communication channels over TLS, where data transfer between master node and worker node are protected using IPSec communication. Protecting data at rest involves the following considerations in relevance to the different types of data to be protected.
   A) **The data required to run the framework such as cloud user credentials.** This data is protected using the MiCADO security component titled Credential Store (refer to Section 3.3 of D7.5 [7] for further details).
   B) **Data related to users responsible for the deployment and management of applications.** Such data is securely managed by another security component of MiCADO framework called Credential Manager and is used through L7 firewall (i.e. Zorp). For further details on these two security components refer to Section 3.4 and 3.5 of D7.5 [7] respectively.
   C) **Application data, i.e. the data created, managed, required by the application.** Such data needs to be also protected while at rest. However, this is a responsibility of the underlying application. Therefore, it is usually managed through a database system, where the MiCADO framework handles deployment and orchestration of the application. As a result, the encryption and protection of application data is *out of the scope* of the MiCADO framework.

   To summarise, MiCADO *fully* satisfies this requirement by protecting data in transit and at rest, except application specific data.
3. ***The system must also run on public cloud: The*** MiCADO framework fully supports the use of a public cloud. Furthermore, MiCADO has been extensively used and tested over the following public clouds: Amazon AWS, Microsoft Azure, CloudSigma and CloudBroker. Hence, the MiCADO framework *fully* satisfies this requirement. The TOSCA-based ADT templates for all these public cloud providers are made available with the open source release of MiCADO and can be found in [9].

Based on the above analysis, the assessment results of the MiCADO framework in accordance to Inycom's use-case requirements are summarised in Table 6.

Table 5 MiCADO assessment in accordance to SAKERs use-case requirement

SAKER use-case requirement	Status	Guidelines
The system must be able to run on a private network or private cloud	Supported	See [10] for deployment guidelines
File encryption may be seen as a requirement depending on the client and application	Partial	
The system must also run on public cloud	Supported	See [10] for deployment guidelines

2.1.3 Outlandish

Outlandish is a cooperative digital agency specialising in middleware, usability, search and scalable data applications. Outlandish mainly focuses is on the interface between computers

and users in insight-generation and data management. Outlandish has considerable experience in building highly usable and intuitive data management solutions. The security requirements obtained from Outlandish are as follows[3]:

1. ***Application secret stores:*** Application secrets are sensitive information that are required to run applications properly. Some examples of such secrets include database account credentials, API keys for third party service, etc. This confidential information is securely handled by Kubernetes during the entire execution of the application. Kubernetes is also responsible for the secure transfer of these secrets to the worker nodes, when required. These secrets are defined by the users through Secret Distribution policies using the TOSCA-based ADT template (refer to Section 3.3.2 in Deliverable D7.4 [6] for further details). Thus, MiCADO *fully* satisfies this requirement.
2. ***Full disk encryption:*** This feature is not further required by Outlandish, therefore, MiCADO framework does not support full disk encryption.
3. ***Black box penetration testing of deployment along with white box security audit:*** The details on the penetration testing of MiCADO framework are presented in Section 3.
4. ***Support for Ansible roles to set up software requirements (for example NGINX and Node.js):*** The MiCADO framework can be deployed through an Ansible playbook. However, this requirement is not related to security of MiCADO framework and therefore, it is not covered in this deliverable. For further details on the deployment of MiCADO through Ansible playbook, please refer to the Deliverable D6.3.

Based on the above analysis, the assessment results of MiCADO framework against the use-case requirements of Outlandish are summarised in Table 7.

Table 6 MiCADO assessment in accordance to Outlandish's use-case requirement

Outlandish use-case requirement	Status	Guideline
Application secret stores	Supported	
Full disk encryption [Not required anymore]	None	
Black box penetration testing of deployment along with white box security audit	Supported	Section 3
Support for Ansible roles to set up software requirements (for example NGINX and Node.js)	Supported	See [10] for deployment guideline and D6.3 deliverable document for details on Ansible playbook.

[3] The implementation of MiCADO framework is analysed and evaluated to reflect on the achievement of these requirements.

2.1.4 CloudSME

CloudSME provides vendor independent cloud technology to support the sustainable growth and digitalization within Europe as well as to increase its competitiveness in the worldwide economy. The security requirements obtained from CloudSME are the following[4].

1. ***TLS support for end-users communicating with a front-end host:*** MiCADO is a generic framework that manages the deployment and the infrastructure required by applications. It has no control and no responsibility to handle internal details of the deployed applications. This requirement, i.e. TLS support for communication of end-users with a front-end host, is specific to applications and therefore, it should be taken care by the actual applications, if required. However, MiCADO can be easily extended to facilitate applications to setup the required TLS support, if required, using the Application Description Template (ADT) based configuration. This will be handled by defining NetworkSecurityPolicy in ADT based configuration to enable TLS communication (refer to Section 3.2, Deliverable 7.4 [6] for further details), as well as utilising application secret mechanism for providing relevant details for SSL certificate/key. However, the latest release of MiCADO (i.e. 0.7.2) does not support the automatic configuration of TLS/ SSL in worker nodes based on defined NetworkSecurityPolicy in ADT template and SSL certificate/ key provisioned by the application secret mechanism. Therefore, this requirement can be considered as partially supported.
2. **Connection within the cluster must be restricted to specific whitelisted IP addresses: The** MiCADO framework relies on the infrastructure details provided by the users themselves at the time of deployment. This let the user to configure any restriction on the underlying cluster of cloud resources prior to deployment. Particularly, the user can configure network policies on the cloud to only allow connections from specific whitelisted IP addresses, then provide such configured infrastructure information in the TOSCA-based ADT at the time of application deployment. Once deployed, MiCADO fully complies with the provided infrastructure and any associated restriction. Hence, connections can be made restricted to specific whitelisted IP address, if desired and configured by the user.

Based on the above analysis, the assessment results of the MiCADO framework against CloudSME use-case requirements are summarised in Table 8.

Table 7 MiCADO assessment in accordance to CloudSME's use-case requirements

CloudSME use-case requirement	Status	Guideline
TLS support for end-users communicating with a front-end host	Partial	
Connection to within the cluster must be restricted to specific whitelisted IP addresses	Partial	

[4] The implementation of MiCADO framework is analysed and evaluated to reflect on the achievement of these requirements.

2.2 Evaluation based on Main Security Requirements

This section discusses and evaluates the MiCADO framework based on its success in fulfilling the main security requirements identified at the start of the project. These security requirements are listed and presented in Deliverable D7.1[3]. Table 8 discusses and analyses the behaviour of the MiCADO framework in respect to each security requirement.

Table 8 MiCADO assessment of the Main Security Requirements

Id	Purpose	Details
SR01	Description	*COLA should provide certain guarantees that the cloud providers that are connected to the service through Cloud Access API are running under trusted state.*
	Assessment	The specification of cloud provider related details is the responsibility of the application owner and MiCADO framework does not have any control over it. Hence there is no mechanism in MiCADO framework to test and validate that the cloud provider Access API are running under trusted state or not.
	Status	Not applicable
SR02	Description	*COLA shall make sure that the cloud providers that are connected to the service through the Cloud Access API are protecting users' data from external attacks by encrypting the entire hard disks of the CSP.*
	Assessment	MiCADO does not have control over the cloud provider's security settings. This depends on the service level agreement (SLA) between application owners and cloud providers. Hence, there is no mechanism in the MiCADO framework ensuring that the CSP encrypts the entire hard disks or not.
	Status	Not applicable.
SR03	Description	*COLA shall guarantee that the credentials of a user can be revoked without affecting the overall performance or the proper function of the service.*
	Assessment	In the MiCADO framework, the credential management and task execution are separate. Hence, credentials revocation does not have any effect on the overall performance. Furthermore, it does not affect the execution of the running service .
	Status	Supported
SR04	Description	*COLA should provide guarantees that all launched VMs are running in a trusted state.*
	Assessment	MiCADO framework launches all VMs on the Cloud provider specified by the user at the time of deployment. As the cloud provider is chosen by users, it implies that the provider is trusted. MiCADO does not include validation mechanism to

		ensure that the VMs shall always run in trusted state.
	Status	Not applicable
SR05	Description	*COLA should enforce TLS communication between all participating instances.*
	Assessment	The following aspects related to MiCADO framework in this regard are important to mention: 1. The components of the MiCADO framework are running inside a security boundary over master node. Therefore, it is not required for the components of the MiCADO framework to communicate amongst each other via TLS. 2. The communication between the master node and worker nodes are secured using the IPsec protocol. 3. The MiCADO framework does not ensure that the application level communication must be TLS supported as this entirely depends on the underlying application. However, it facilitates applications to setup the TLS support, if required, using the Application Description Template (ADT) based configuration. This will be handled by defining NetworkSecurityPolicy in ADT based configuration to enable TLS communication (refer to Section 3.2, Deliverable 7.4 [6] for further details), as well as utilising application secret mechanism for providing relevant details for TLS certificate/key. However, the MiCADO release (0.7.2) at the time of writing does not support the automatic configuration of TLS in worker nodes based on defined NetworkSecurityPolicy in ADT template and TLS certificate/key provisioned by the application secret mechanism. 4. The communication towards MiCADO master node (for instance, submitting ADT to MiCADO) is protected under TLS protocol.
	Status	Partial
SR06	Description	*COLA shall not be operational if TLS is terminated (e.g. recognise TLS stripping techniques).*
	Assessment	Based on the description provided in response to the **SR05** requirement, it is not mandatory to enforce TLS communication. It depends on the requirements of the underlying application. The MiCADO framework does not have any mechanism to enforce TLS communication required for the application; this should be handled by the application.
	Status	Not applicable
SR07	Description	*COLA should allow entities with certain access rights to*

		define certain security profiles that will considered as trusted.
	Assessment	The MiCADO framework facilitates application owners to define the application's security requirements, namely secrets and firewall rules in the application TOSCA descriptor.
	Status	Supported
SR08	Description	*COLA should use/propose a mechanism that protects sensitive data that are temporarily stored in the memory.*
	Assessment	In the context of MiCADO framework, the 3rd party components, like HashiCorp Vault, Kubernetes and etcd, managed by the Kubernetes cluster, store secrets in the memory as encrypted. The MiCADO components running on the master node are not required to encrypt the secrets in temporary memory storage because all MiCADO components run inside a security boundary on the master node. However, MiCADO does not implement a mechanism to enforce user applications to store secrets as encrypted in the memory.
	Status	Partial
SR09	Description	*COLA should ensure that deployed applications in the application server are trusted using a mature trust model.*
	Assessment	It is not possible for the MiCADO framework to ensure that the deployed applications on the application server must be trusted using a mature trust model. This is an application specific task and therefore, this assurance must be taken care of by the applications.
	Status	Not applicable.
SR10	Description	*COLA shall be operational only if all security components are active.*
	Assessment	The MiCADO framework is only fully operational when all security components including Zorp firewall, IPSec, SPM and Vault are functional. However, the Image Integrity Verifier (IIVR) is not required to be active. In the case of the failure of any of these required security components during operation, then the MiCADO framework will also stop working. E.g. if communication to any of the worker nodes fails due to IPsec or firewall failure or any other connectivity issue, then Occopus as a result will terminate the VM in the cloud. The changes in infrastructure will be reflected on Kubernetes dashboard. If other security components fail then the master node will cease to operate.
	Status	Supported
SR11	Description	*COLA should provide a proper and efficient mechanism for key revocation of the misbehaving entities.*
	Assessment	All MiCADO components/entities run inside a security

		boundary on the master node and do not rely on any key distribution and revocation mechanism. The communication between the master node and MiCADO users outside the security boundary, is protected over TLS. Communication between the master node and worker nodes is protected over IPsec. TLS and IPsec work based on revocable X.509 certificates .
	Status	Supported
SR12	Description	*COLA shall use a key generation algorithm that guarantees that the generated keys are secure (i.e. long enough) and that secret keys are not statically stored in one place (e.g. on the application server or in the application).*
	Assessment	For TLS, a 4096-bit RSA key is generated. For IPsec, a minimum of 2048 bits for RSA keys is enforced. Both key lengths are secure enough. On the other hand, since all components inside the security boundary of the master node are trusted, there is no need to store the keys in separate places. Keys are stored inside the master node. The Credential Store component inside the master node can be utilized to protect such keys at rest.
	Status	Supported
SR13	Description	*COLA should guarantee that when a user's key is compromised the rest of the keys must not be revoked.*
	Assessment	The MiCADO users are not granted any keys. However, all users have their username and password. In case, if a user's password is compromised, only her password needs to be revoked/reset without affecting other users' passwords. Hence, the password reset mechanism is supported.
	Status	Supported

Table 9 MiCADO assessment of Cloud Compute Security Requirements (CCSR)

Id	Purpose	Details
CCSR-1	Description	*COLA should provide mechanisms to enforce secure destruction of workloads and configuration.*
	Assessment	MiCADO send requests to destroy the running VMs when not required. Upon reception, the cloud provider destroys the VMs. However, the secure destruction of VM (as well as workloads and configuration inside the VM) is enforced by cloud providers, which MiCADO does not provide mechanisms to verify it.
	Status	None
CCSR-2	Description	*COLA should provide mechanisms for placement selection of workloads. Tenants may put forth requirements towards the*

		placement of workloads according to pre-defined criteria, e.g. geographical, jurisdictional or administrative placement criteria.
	Assessment	In the MiCADO framework, application owners have to provide infrastructure related details and policies using an easy-to-use TOSCA-based ADT at the time of deployment. This implies that application owners have to choose the infrastructure by themselves and MiCADO framework complies with these configurations for the entire execution time. The MiCADO framework does not enforce any restriction with respect to locality-based workload placement as it is explicit to the user and the cloud service provider.
	Status	Not applicable.

Table 10 MiCADO assessment of Cloud Storage Security Requirements (CSSR)

Id	Purpose	Details
CSSR-1	Description	*COLA should provide mechanisms to enforce secure destruction of data upon storage decommissioning.*
	Assessment	In case data storage is initialized in a VM by MiCADO, based on the user's demand described in ADT, it will be deleted along with the VM's destruction. However, the destruction and its security are enforced by cloud providers. MiCADO does not provide a mechanism to verify that it has happened .
	Status	None
CSSR-2	Description	*COLA should provide mechanisms for placement selection of data. Tenants may put forth requirements towards the placement of data storage according to pre-defined criteria, e.g. geographical, jurisdictional or administrative placement criteria.*
	Assessment	Similar to the assessment description in response of **CCSR-1**, MiCADO framework does not enforce any restriction with respect to locality-based data storage as it is explicit to the user and the cloud service provider.
	Status	Not applicable.

Table 11 MiCADO assessment as per Cloud Network Security Requirements (CNSR)

Id	Purpose	Details
CNSR-1	Description	*COLA should provide support access control for cloud network infrastructure.*
	Assessment	MiCADO is responsible for access control to network software inside MiCADO such as Zorp firewall. This is handled by facilitating administrative access to MiCADO master node through SSH to access the components, e.g. (Zorp firewall in this case) of MiCADO framework. On the other hand, access control support to the cloud network infrastructure depends on the chosen cloud providers.
	Status	Partial

CNSR-2	Description	*COLA should provide support verification of deployed network configuration policies.*
	Assessment	MiCADO currently does not support verification of deployed network configuration policies.
	Status	None
CNSR-3	Description	*COLA should provide support mechanisms for authentication, confidentiality and integrity protection of network infrastructure.*
	Assessment	Authentication, confidentiality and integrity are provided through TLS and IPsec protocol in the MiCADO framework.
	Status	Supported
CNSR-4	Description	*COLA should provide support traceability of network infrastructure management.*
	Assessment	Currently, MiCADO does not provide any feature that support traceability of network infrastructure management.
	Status	None
CNSR-5	Description	*COLA should provide support isolation of tenant policy domains.*
	Assessment	MiCADO currently supports single tenant; therefore, there is no need to isolate tenant policy domains.
	Status	Not applicable
CNSR-6	Description	*COLA shall verify submitted network configuration and management policies prior to deployment.*
	Assessment	MiCADO currently does not support verifying the submitted network configuration and management policies prior to deployment.
	Status	None
CNSR-7	Description	*COLA shall enforce tenant quota isolation.*
	Assessment	MiCADO supports deploying applications for single tenant Therefore, there is no need to enforce tenant quota isolation inside MiCADO.
	Status	Not applicable
CNSR-8	Description	*COLA shall support authentication of endpoints enrolled into the network infrastructure.*
	Assessment	TLS provides endpoint authentication in which MiCADO master is authenticated to MiCADO users. Furthermore, IPsec ensures authentication between the master node and worker nodes in order to secure their communication.
	Status	Supported
CNSR-9	Description	*COLA shall support deployment of secure communication channels for in-transit protection of data among the endpoints of the network infrastructure.*
	Assessment	Data communicated between MiCADO users and the master node is protected under TLS. In addition to that, data transferred among the master node and worker nodes is secure through IPsec protocol.
	Status	Supported
CNSR-10	Description	*COLA shall support limiting access to the network infrastructure according to pre-defined network properties of*

		the endpoints.
	Assessment	On the master node, only necessary ports to its internal components are opened. On the worker nodes, only necessary ports for communicating with the master node, and ports for applications which are defined in ADT are open. Access to the framework is enforced to the pre-defined restriction provided in the ADT by the application owner at the time of deployment.
	Status	Supported

2.3 Evaluation based on the NIST 800-53 compliance control reference

This section evaluates the MiCADO framework on the integration and usage of the well-established industrial standards NIST SP 800-53 [8]. The NIST SP 800-53 is a set of standards and guidelines, created to enhance the security and privacy of information systems used within the United States federal government. The NIST SP 800-53 introduces a list of security controls that facilitates the development of robustly secure and resilient information systems. The provided controls span across operational, technical, and management safeguards that are required to be used by information systems to maintain the security and integrity of United States federal information systems. These controls are classified into 20 different families.

The following paragraphs discusses the use of NIST SP 800-53 (or relevant) controls by the MiCADO framework. In addition to that, the assessment of MiCADO framework, against each activity of the relevant control, is summarized in respective tables.

Please note that the abbreviations of each family of controls and the individual controls, used in the following description, are following the official documentation of NIST SP 800-53. The key purpose of mentioning these abbreviations is to avoid any inconsistency and for referencing purposes, if required.

2.3.1 Access Control (AC)

This family of controls consists of many controls related to the management of access to the underlying system. However, only the following limited number of key security related controls of this family are discussed based on its relevancy to the underlying system, i.e. MiCADO:

1. *AC-2 Account management:* This control is concerned with the management of users' accounts. In the context of MiCADO framework, users' accounts include definition of users and their roles. In the current release (i.e. 0.7.2), MiCADO supports two roles (i.e. user and admin) and the access control policy is quite simple (see Table 9). However, further roles as well as policy extension can be easily supported.

Table 12 Role-based access control in MiCADO

Role	Permission
User	Access the Dashboard
admin	Access the Dashboard, Submit application to the framework

MiCADO framework facilitate users and their role management through a security component called Credential Manager. MiCADO framework empowers the user, responsible for the deployment, to create an admin user at the time of deployment. Once MiCADO framework is deployed then the admin user has the power, through a command line utility, to create further users and assign them roles. For complete details on the features of the Credential Manager refer to Section 3.4 of deliverable D7.5 [7]. To assess the MiCADO framework in the context of this control, the MiCADO framework is evaluated against all the defined activities under this control in the NIST standard document [8]. Table 10 lists all these activities, their brief description, corresponding MiCADO assessment, implementation status and relevant guidelines.

Table 13 MiCADO assessment in accordance to AC-2 control

Activities	Purpose	Details
AC2-1	Description	*Define and document the types of system accounts.*
	Assessment	Currently MiCADO supports two types of system accounts: 'admin' and 'user'. These system account types reflect user roles and they are extendable.
	Status	Supported
	Guidelines	GL2, GL3, GL5, GL6
AC2-2	Description	*Assign account manager for system accounts*
	Assessment	MiCADO does not explicitly assign an account manager for system accounts. Instead, an account manager needs SSH key to access the Master Node, and then use the command line facilities to manage users.
	Status	Supported
	Guidelines	GL4
AC2-3	Description	*Establish conditions for group and role membership*
	Assessment	MiCADO only supports the role membership, which is configurable by the application owner.
	Status	Partial
	Guidelines	GL6, GL2, GL3, GL5
AC2-4	Description	*Specifies authorized users of the information system, group and role membership, and access authorizations (i.e., privileges) and other attributes (as required) for each account.*
	Assessment	Role-based access authorizations (privileges) are configurable during deployment (please refer to GL6 in sub-section 2.3.6 as a guideline for configuring role-based access control).
	Status	Supported
	Guidelines	GL6
AC2-5	Description	*Requires approvals by [Assignment: organization-defined personnel or roles] for requests to create information system accounts*

	Assessment	It is not in the scope of MiCADO framework, and is directly under the control of application owners.
	Status	Not Applicable
	Guidelines	
AC2-6	Description	*Creates, enables, modifies, disables, and removes information system accounts in accordance with [Assignment: organization-defined procedures or conditions].*
	Assessment	It is under the control of application owners. However, MiCADO supports command line facilities to create/ modify/ remove system accounts.
	Status	Not applicable
	Guidelines	
AC2-7	Description	*Monitors the use of information system accounts.*
	Assessment	MiCADO does not support such a feature.
	Status	Not available
	Guidelines	
AC2-8	Description	*Notifies account managers: (1) When accounts are no longer required; (2) When users are terminated or transferred; and (3) When individual information system usage or need-to-know changes.*
	Assessment	MiCADO does not support such a feature.
	Status	Not available
	Guidelines	
AC2-9	Description	*Authorizes access to the information system based on: (1) A valid access authorization; (2) Intended system usage; and (3) Other attributes as required.*
	Assessment	MiCADO supports role-based authorization.
	Status	Supported
	Guidelines	GL3, GL5, GL6
AC2-10	Description	*Reviews accounts for compliance with account management requirements [Assignment: organization-defined frequency].*
	Assessment	It is out of the scope of MiCADO framework.
	Status	None
	Guidelines	
AC2-11	Description	*Establishes a process for reissuing shared/group account credentials (if deployed) when individuals are removed from the group.*
	Assessment	MiCADO does not support group account credentials.
	Status	None
	Guidelines	

2. *AC-3 Access enforcement:* This control deals with authorizations of logical access to information and system resources in accordance to the applicable access control policies.

The MiCADO framework in itself is *not* a typical information system, but an orchestration solution that facilitates the management and deployment of applications over the cloud. Therefore, MiCADO does not handle the application related authorization access. However, MiCADO facilitates the use of multiple users that can handle the management of the underlying application. For this purpose, MiCADO uses the following two different security components: *(1)* Credential Manager as a utility to create and manage users and their roles, and *(2)* L7 firewall (i.e. Zorp) to handle and enforce valid access to system by mediating between user login and credential manager; hence, facilitating access to system for valid users only. Table 11 lists the activity of this control, current status and the relevant guideline.

Table 14 MiCADO assessment in accordance to AC-3 control

Activities	Purpose	Details
AC-3	Description	*Enforce approved authorizations for logical access to information and system resources in accordance with applicable access control policies.*
	Assessment	MiCADO, at the time of initial deployment, only empowers the admin user to access the system resources and other management activities, i.e. deployment (and un-deployment) of the application. The admin at a later stage can create other users and assign them different roles. MiCADO at any stage enforces the access of the users to the system based on their assigned roles.
	Status	Supported
	Guidelines	GL3, GL5

3. *AC-7 Unsuccessful logins attempts:* MiCADO currently does not support this feature to lock the account for certain time after unsuccessful login attempts. However, the design of Credential Manager security component can be enhanced and support such a feature – something that is considered for implementation and integration in future releases. Table 12 lists the activities of this control.

Table 15 MiCADO assessment in accordance to AC-7 control

Activities	Purpose	Details
AC7-1	Description	*Enforces a limit of [Assignment: organization-defined number] consecutive invalid logon attempts by a user during a [Assignment: organization-defined time period].*
	Assessment	
	Status	Not available
	Guidelines	
AC7-2	Description	*Automatically [Selection: locks the account/node for an [Assignment: organization-defined time period]; locks the account/node until released by an administrator; delays next logon prompt according to [Assignment: organization-defined delay algorithm]] when the maximum number of unsuccessful attempts*

		is exceeded.
	Assessment	
	Status	Not available
	Guidelines	

4. *Notifications:* The NIST 800-53 Access Control family contains the following two controls regarding notifications: (1) *AC-8 System use*, (2) and *AC-9 Previous Log-on (Access) notifications*. Some examples of such notifications include unauthorized access, unsuccessful login attempts, change information, e.t.c. the MiCADO framework currently does not support any such notifications. However, MiCADO Credential Manager security component facilitates some aspects of user management related notifications; such as password change, user account change details, unsuccessful login attempts. Please note that these notifications are not enabled in the current release, however, and are considered as a future addition. Table 13 summarizes the assessment of the MiCADO framework in accordance to the activities of this control.

Table 16 MiCADO assessment in accordance to AC-8 and AC-9 controls

Activities	Purpose	Details
AC8-1	Description	*Displays [Assignment: organization-defined system use notification message or banner] to users before granting access to the system that provides privacy and security notices consistent with applicable U.S. federal laws, Executive Orders, directives, policies, regulations, standards, and guidance and states that: (1) Users are accessing a U.S. Government system; (2) System usage may be monitored, recorded, and subject to audit; (3) Unauthorized use of the system is prohibited and subject to criminal and civil penalties; and (4) Use of the system indicates consent to monitoring and recording*
	Assessment	MiCADO is a framework that can be used by an organization or an individual to run their applications. It is under full control of such organizations/ individuals. Therefore, there is no need to display a notification to users before granting access to the framework.
	Status	None
AC8-2	Description	*Retains the notification message or banner on the screen until users acknowledge the usage conditions and take explicit actions to log on to or further access the information system*
	Assessment	MiCADO is not an information system and under full control of the application owners, hence no need to acknowledge users before letting them log into the framework.
	Status	None
AC8-3	Description	*For publicly accessible systems: (1) Displays system use information [Assignment: organization-defined conditions], before granting further access; (2) Displays references, if any, to monitoring, recording, or auditing that are consistent with privacy*

		accommodations for such systems that generally prohibit those activities; and (3) Includes a description of the authorized uses of the system
	Assessment	MiCADO framework is fully controlled by the application owners, hence no need to display information about system use.
	Status	None
AC9	Description	*Notifies the user, upon successful logon (access) to the system, of the date and time of the last logon (access).*
	Assessment	This feature is not available in the current release (0.7.2).
	Status	Not available

5. *AC-12 Session termination:* The L7 firewall security component of MiCADO framework (i.e. Zorp) handles the user session and terminates it in the case of inactivity timeout.

2.3.2 Audit and Accountability (AU)

This family consists of controls related to security auditing of underlying system. It includes security controls like AU-2 Audit events, AU-3 Content of audit records, etc. As per the NIST 800-53 supplemental guidance, the organization should identify the audit events types that are important and relevant to the security of the system. Some examples of such events are password changes, failed login attempts, security attribute changes, etc. Currently, MiCADO framework does not support auditing events.

2.3.3 Awareness and Training (AT)

This family consists of controls related to security awareness and training such as *AT-1 Security and awareness training policy, AT-2 Awareness training, AT-3 Role-based training*, etc. In this regard, MiCADO provides guidelines (refer Section 2.3 for further details) related to various Security and privacy related configurations of MiCADO. Furthermore, these guidelines will be made available freely with the future public releases of MiCADO. The following table lists the related activities of this control in relevance to MiCADO framework.

Table 17 MiCADO assessment in accordance to AT controls

Activities	Purpose	Details
AT1-1	Description	*Develop, document, and disseminate to [Assignment: organization-defined personnel or roles]: (1) A security and privacy awareness and training policy that: (a) Addresses purpose, scope, roles, responsibilities, management commitment, coordination among organizational entities, and compliance; and (b) Is consistent with applicable laws, Executive Orders, directives, regulations, policies, standards, and guidelines; and (2) Procedures to facilitate the implementation of the security and privacy awareness and training policy and the associated security and privacy awareness and training controls.*
	Assessment	MiCADO security features are fully documented in deliverables D7.1, D7.2, D7.3, D7.4, D7.5 [3][4][5][6][7]. The guidelines for

		configuring such features are provided in Section 2.3.
	Status	Supported
AT1-2	Description	*Designate an [Assignment: organization-defined senior management official] to manage the security and privacy awareness and training policy and procedures*
	Assessment	This activity relies on organizations; therefore, it is out of the scope of the MiCADO framework.
	Status	None
AT1-3	Description	*Review and update the current security and privacy awareness and training: (1) Policy [Assignment: organization-defined frequency]; and (2) Procedures [Assignment: organization-defined frequency]*
	Assessment	The current security configuration guidelines are based upon the latest release of MiCADO (0.7.2). It shall be updated for future releases.
	Status	Supported
AT1-4	Description	*Ensure that the security and privacy awareness and training procedures implement the security and privacy awareness and training policy and controls.*
	Assessment	The guidelines related to the security and privacy configuration of the MiCADO framework is provided in this document. These guidelines will be also available with the public release of the MiCADO framework. On the other hand, the security and privacy awareness of the underlying application are the responsibility of the organisation, which is out of the scope of the MiCADO framework.
	Status	Partial
AT1-5	Description	*Develop, document, and implement remediation actions for violations of the awareness and training policy.*
	Assessment	This activity relies on organizations; therefore, it is out of scope of MiCADO.
	Status	None
AT2	Description	*Provide basic security and privacy awareness training to system users (including managers, senior executives, and contractors): (a) As part of initial training for new users; (b) When required by system changes; and (c) [Assignment: organization-defined frequency] thereafter.*
	Assessment	This activity relies on organizations; therefore, it is out of the scope of MiCADO. However, regarding the security and privacy of the MiCADO framework, further training to system users will be provided, if required.
	Status	None
AT3	Description	*Provide role-based security and privacy training to personnel*

		with the following roles and responsibilities: [Assignment: organization-defined roles and responsibilities]: (a) Before authorizing access to the system or performing assigned duties; (b) When required by system changes; and (c) [Assignment: organization-defined frequency] thereafter.
	Assessment	This activity relies on organizations; therefore, it is out of scope of MiCADO.
	Status	None

2.3.4 Assessment, authorization and monitoring (CA)

The key security related control from this family is the CA-8 Penetration testing. As per the supplemental guidelines of NIST 800-53, Penetration testing can be used to either validate vulnerabilities or determine the degree of penetration resistance of systems to adversaries within specified constraints. In the context of MiCADO framework, penetration testing has been performed and the results are reported in Section 3.

2.3.5 Configuration management (CM)

The key security related controls in the context of MiCADO framework includes *CM-3 (Configuration Change Control), CM-3 [6] (Cryptography Management), CM-5 (Access restriction for change), CM-9 (Configuration management plan)* and *CM-11 (User installed software).* The MiCADO framework facilitates the deployment and management of an application through fully configurable settings provided in a TOSCA-based ADT file. These configurable settings include aspects related to the application and the underlying infrastructure such as repository details of docker containers, reference to a virtual machine image, communicating port details, scalability policies, e.t.c . C hanges to these configurations after deployment is also possible. These changes are restricted to specific user, who has the privilege to handle application deployment. However, this history of changes is not recorded to be viewed at a later stage or for any auditing purposes.
In terms of Cryptography management, MiCADO supports and facilitates TLS based communication between user and Master node. In this regard, MiCADO empowers users to specify a certificate and key to enable TLS based communication. In terms of *User installed software*, MiCADO framework does not directly install or request any configuration for the software/utilities required for application execution on the worker nodes. In this respect , MiCADO relies on the VM (and/or container) images, which are specified by the application owner at the time of deployment. Therefore, if an application relies on any dependencies, it must be dealt with beforehand in the provided images for VM (and/or container). Additionally, any freely available dependencies can be also specified in the TOSCA file at the time of deployment. Hence, the responsibility of installing such software and/or any associated issues laid on application owner rather than MiCADO framework. Table 17 lists the activities and their assessment status.

Table 18 MiCADO assessment in accordance to CM controls

Activities	Purpose	Details
CM3-1	Description	*Determine the types of changes to the system that are configuration-controlled.*

	Assessment	The guidelines in Section 2.3 contains configurable changes to MiCADO.
	Status	Supported
	Guidelines	GL1, GL6
CM3-2	Description	*Review proposed configuration-controlled changes to the system and approve or disapprove such changes with explicit consideration for security impact analyses*
	Assessment	A team of security experts performed the security impact analyses of any change/enhancement required before deciding of its approval or disapproval. Furthermore, the introduction of a formal process is aimed. Such a process will deal and document any proposed changes/enhancements.
	Status	Partial
	Guidelines	
CM3-3	Description	*Document configuration change decisions associated with the system: During development, any changes have been reviewed and discussed with consideration for security impact.*
	Assessment	The changes have been documented in the previous deliverables. However, these changes cannot be systematically tracked.
	Status	Partial
	Guidelines	
CM3-4	Description	*Implement approved configuration-controlled changes to the system*
	Assessment	Any changes once approved are implemented in the next release. All changes (and improvements) with respect to different releases can be seen from Github repository [12].
	Status	Supported
	Guidelines	
CM3-5	Description	*Retain records of configuration-controlled changes to the system for [Assignment: organization-defined time-period]*
	Assessment	All the various changes and improvements proposed during the development stages are documented in the previous deliverables. In addition, the changes/enhancement from one release to another can also be seen from Github repository [12].
	Status	Supported
	Guidelines	
CM3-6	Description	*Monitor and review activities associated with configuration-controlled changes to the system.*
	Assessment	It is not supported yet.
	Status	Not available
	Guidelines	
CM3-7	Description	*Coordinate and provide oversight for configuration change control activities through [Assignment: organization-defined configuration change control element] that convenes [Selection (one or more): [Assignment: organization-defined frequency]; [Assignment: organization-defined configuration change conditions]].*
	Assessment	

	Status	Not available
	Guidelines	
CM-3[6]	Description	*Ensure that cryptographic mechanisms used to provide [Assignment: organization-defined security safeguards] are under configuration management.*
	Assessment	Users are able to specify certificate and key to enable TLS based communication.
	Status	Supported
	Guidelines	GL1
CM-5	Description	*Define, document, approve, and enforce physical and logical access restrictions associated with changes to the system.*
	Assessment	
	Status	Not available.
	Guidelines	
CM-9-1	Description	*Addresses roles, responsibilities, and configuration management processes and procedures*
	Assessment	
	Status	Not available.
	Guidelines	
CM-9-2	Description	*Establishes a process for identifying configuration items throughout the system development life cycle and for managing the configuration of the configuration items.*
	Assessment	
	Status	Not available
	Guidelines	
CM-9-3	Description	*Defines the configuration items for the system and places the configuration items under configuration management.*
	Assessment	
	Status	Not available
	Guidelines	
CM-9-4	Description	*Is reviewed and approved by [Assignment: organization-defined personnel or roles].*
	Assessment	
	Status	Not available.
	Guidelines	
CM-9-5	Description	*Protects the configuration management plan from unauthorized disclosure and modification.*
	Assessment	
	Status	Not available.
	Guidelines	
CM-11-1	Description	*Establish [Assignment: organization-defined policies] governing the installation of software by users:*
	Assessment	It is under full control of the application owners and out of the scope of MiCADO framework.
	Status	None
	Guidelines	
CM-11-2	Description	*Enforce software installation policies through the following methods: [Assignment: organization-defined methods]*

	Assessment	It is out of the scope of MiCADO framework.
	Status	None
	Guidelines	
CM-11-3	Description	*Monitor policy compliance at [Assignment: organization-defined frequency]*.
	Assessment	This activity is related to information systems and is not relevant to the MiCADO framework.
	Status	None
	Guidelines	

2.3.6 Identification and authentication (IA)

The key security related controls, in the context of MiCADO framework, from this family includes *IA-2 Identification and authentication [organizational users], IA-5[1] Password based authentication, IA-8 Identification and authentication [Non-organizational users]*. In the MiCADO framework, the user, who has access to the MiCADO master node, can create further users with different roles. These users are then allowed to use the MiCADO framework as per their role description. However, no records of users' association with an organization are managed by the MiCADO framework. Hence, all defined users have access to the MiCADO framework and will be authenticated before use. The MiCADO framework currently only supports username/password-based authentication. The current authentication is handled by a L7 firewall (i.e. Zorp) and the Credential Manager component of the MiCADO framework. It also enforces various password related guidelines such as storing passwords using an approved hash algorithm, minimum characters length, mix upper/lower-case letters, special characters, e.t.c. Table 18 lists the activities and their assessment status.

Table 19 MiCADO assessment in accordance to IA controls

Activities	Purpose	Details
IA-2	Description	*Uniquely identify and authenticate organizational users or processes acting on behalf of organizational users:*
	Assessment	Each user is uniquely identified in the MiCADO framework by user name.
	Status	Supported
	Guidelines	GL3, GL5
IA5-1	Description	*Verifying, as part of the initial authenticator distribution, the identity of the individual, group, role, or device receiving the authenticator*:
	Assessment	
	Status	Not available
	Guidelines	
IA5-2	Description	*Establishing initial authenticator content for any authenticators issued by the organization*
	Assessment	MiCADO supports the functionality to create a user with initial authenticator, (i.e. password) content which is provided as input.
	Status	Supported
	Guidelines	GL4
IA5-3	Description	*Ensuring that authenticators have sufficient strength of*

		mechanism for their intended use.
	Assessment	The authenticator (i.e. password) is enforced to follow pre-defined rules defined in the Credential Manager component.
	Status	Supported
	Guidelines	
IA5-4	Description	*Establishing and implementing administrative procedures for initial authenticator distribution, for lost/compromised or damaged authenticators, and for revoking authenticators.*
	Assessment	
	Status	Not available.
	Guidelines	
IA5-5	Description	*Establishing minimum and maximum lifetime restrictions and reuse conditions for authenticators*
	Assessment	
	Status	Not available.
	Guidelines	
IA5-6	Description	*Changing/refreshing authenticators [Assignment: organization-defined time-period by authenticator type].*
	Assessment	Authenticator changes happen as required instead of in a defined time-period.
	Status	Not available
	Guidelines	
IA5-7	Description	*Protecting authenticator content from unauthorized disclosure and modification.*
	Assessment	Authenticators (i.e. passwords) are stored in a secure manner, i.e. hash values instead of plain text.
	Status	Supported
	Guidelines	
IA5-8	Description	*Requiring individuals to take, and having devices implement, specific security controls to protect authenticators.*
	Assessment	Out of the scope.
	Status	None
	Guidelines	
IA5-9	Description	*Changing authenticators for group/role accounts when membership to those accounts changes*
	Assessment	MiCADO does not provides group membership.
	Status	None.
	Guidelines	
IA5[1]-1	Description	*Maintain a list of commonly-used, expected, or compromised passwords and update the list [Assignment: organization-defined frequency] and when organizational passwords are suspected to have been compromised directly or indirectly.*
	Assessment	
	Status	Not available
	Guidelines	
IA5[1]-2	Description	*Verify, when users create or update passwords, that the*

		passwords are not found on the organization-defined list of commonly-used, expected, or compromised passwords.
	Assessment	
	Status	Not available
	Guidelines	
IA5[1]-3	Description	*Transmit only cryptographically-protected passwords:*
	Assessment	The passwords are protected over TLS communication
	Status	Supported
	Guidelines	
IA5[1]-4	Description	*Store passwords using an approved hash algorithm and salt, preferably using a keyed hash.*
	Assessment	The passwords are hashed by facilitating the function *hash_password* from Flask-User package [11], which is customizable to utilize various approved hash algorithm such as SHA512 HMAC, etc.
	Status	Supported
	Guidelines	
IA5[1]-5	Description	*Require immediate selection of a new password upon account recovery.*
	Assessment	
	Status	Not available
	Guidelines	
IA5[1]-6	Description	*Allow user selection of long passwords and passphrases, including spaces and all printable characters.*
	Assessment	A password policy is configurable through regular expression in the Credential Manager component.
	Status	Supported
	Guidelines	
IA5[1]-7	Description	*Employ automated tools to assist the user in selecting strong password authenticators.*
	Assessment	
	Status	Not available
	Guidelines	
IA-8	Description	*Uniquely identify and authenticate non-organizational users or processes acting on behalf of non-organizational users.*
	Assessment	MiCADO does not differentiate between organizational and non-organizational users. Instead, each organization launches and manages their own instance of the MiCADO framework.
	Status	Supported
	Guidelines	

2.3.7 Individual Participation (IP)

The key security related controls, in the context of MiCADO framework, from this family includes *IP-3 Redress* and *IP-6 Individual access*. These controls deal with providing users to access their personally identifiable information and amend, if required. The credential manager component of MiCADO securely stores users' information and facilitates users to

view and modify them, if required. Hence the MiCADO framework satisfy both these requirements. Table 19 lists the activities of this control family and their assessment.

Table 20 MiCADO assessment in accordance to IP controls

Activities	Purpose	Details
IP3-1	Description	*Establish and implement a process for individuals to have inaccurate personally identifiable information maintained by the organization corrected or amended.*
	Assessment	Amendments are handled through the command line facilities in the Master Node.
	Status	Supported
	Guidelines	GL4
IP3-2	Description	*Establish and implement a process for disseminating corrections or amendments of personally identifiable information to other authorized users of the personally identifiable information.*
	Assessment	In the context of the MiCADO framework, the personally identifiable information is relevant to each user independently; therefore, there's no need for disseminating such information to other authorized users.
	Status	None
	Guidelines	
IP6	Description	*Provide individuals the ability to access their personally identifiable information maintained in organizational systems of records.*
	Assessment	MiCADO currently provides a command line facility on the Master Node to view user information.
	Status	Supported
	Guidelines	GL4

2.3.8 Planning (PL)

The key security related controls, in the context of the MiCADO framework, from this family includes *PL-9 Security and privacy architecture*. The MiCADO security architecture adopts a centrally managed modular approach and consists of a number of security components. The full details of these components and the architecture model can be found in D7.2 deliverable document [4]. The design of these MiCADO security components satisfy the different types of requirements that have been formulated during the requirements analysis and adversarial model introduced in D7.1 deliverable document [3].

2.3.9 System and Communications Protection (SC)

The key security related controls, in the context of MiCADO framework from this family includes *SC-3 Security function isolation*, *SC-13 Cryptographic protection*, *SC-17 Public key infrastructure certificates*, and *SC-28 Protection of information at rest*. The key characteristic of the MiCADO framework is its underlying, loosely coupled, interconnecting components-based architecture, where each component is responsible for an independent set of functionalities. This generic approach is also adhered to by all security related components of

the framework. All security component (e.g. L7 firewall or Zorp, Integrity Verifier, Credential Manager, etc) are responsible to perform different functions and the underlying implementation details of each component is not important for other components. Furthermore, all non-security components are not responsible to fulfil any security requirements. Hence, the modular design of MiCADO framework fully isolate the security-oriented functions from the main task of MiCADO framework, i.e. orchestration and management of system resources. In terms of Cryptographic protection and public key infrastructure certificates, MiCADO supports and facilitates TLS based communication between user and Master node. In this regard, MiCADO empowers users to specify certificate and key to enable TLS based communication. Lastly, the 2nd paragraph of Section 2.1.4 discusses in details the protection of information at rest. Table 18 further lists the activities of this control family and their assessment results.

Table 21 MiCADO assessment in accordance to SC controls

Activities	Purpose	Details
SC-3	Description	*Isolate security functions from non-security functions*
	Assessment	The design of MiCADO is modular, that help to separate security components from other components.
	Status	Supported
	Guidelines	
SC-13	Description	*Implement the following cryptographic uses and type of cryptography for each use: [Assignment: organization-defined cryptographic uses and type of cryptography required for each use].*
	Assessment	MiCADO is following cryptographic uses such as hashing for stored password, encryption for data in transit, etc. The framework is provided as open source, which could be customized to follow any organization-defined type of cryptography required for each use.
	Status	Supported
	Guidelines	
SC-17	Description	*Issue public key certificates under an [Assignment: organization-defined certificate policy] or obtain public key certificates from an approved service provider:*
	Assessment	The current release of MiCADO (0.7.2) only supports issuing self-signed public key certificates for setting up TLS communication between users and the Master Node.
	Status	Not available
	Guidelines	
SC-28	Description	*Protect the [Selection (one or more): confidentiality; integrity] of [Assignment: organization-defined information] at rest:*
	Assessment	The Credential Store component of the MiCADO framework securely stores the sensitive information.
	Status	Supported
	Guidelines	

2.4 Security configuration guidelines

The previous section discussed the assessment of the MiCADO framework in relevance to the security requirements obtained from the project's use-cases as well as by analysing well-established standards and guidelines from NIST. Most of the MiCADO supported security features are configurable and therefore, users, who are responsible for deploying their application onto the framework, are supposed to configure them properly before utilization. In this regard, this section aims to provide some guidelines for the configuration of the various security related aspects of the MiCADO framework.

2.4.1 GL1 - TLS configuration

MiCADO currently supports *self-signed* and *user-supplied* TLS configuration. In the case of self-signed certificates, MiCADO generates the certificate itself, whereas in the user-supplied case, the user is responsible for providing the certificate and key file for setting up TLS. Further support such as requesting certificates directly through MiCADO from a third-party provider, such as Let's Encrypt, is considered for future development. The following instructions are provided in reference to the case, when a user launches the MICADO framework through the provided Ansible playbook (refer to [10] for further details on deployment instruction). In order to utilize self-signed TLS, users should edit the file '*ansible-micado/credentials-micado.yml*' as below:

Table 22 Configuring self-signed certificate for TLS

```
tls:
    provision_method: self-signed
```

Alternatively, in the case of user supplied certificate, the users must indicate the provision method as self-supplied and will provide certificate and key files as following:

Table 23 Configuring user-supplied certificate for TLS

```
tls:
  provision_method: user-supplied

  # these need to be set if user-supplied is chosen
  cert: |
-----BEGIN CERTIFICATE-----
MIIFtDCCA5ygAwIBAgIIN29EMh1zHKUwDQYJKoZIhvcNAQEFBQAwMjEXMBUGA1UE
AwwOdnBuLmJhbGFiaXQuaHUxFzAVBgNVBAoMDkJhbGFiaXQgSVQgS2Z0MB4XDTEz
MDYwNjA5MTg0M1oXDTE1MDYwNjA5MTg0M1owgYAxHzAdBgkqhkiG9w0BCQEWEGJs
aW50QGJhbGFiaXQuaHUxFjAUBgNVBAMMDUtvdmFjcyBCYWxpbnQxCzAJBgNVBAsM
AklUMRgwFgYDVQQKDA9CYWxhYml0IElUIEtmdC4xETAPBgNVBAcMCEJ1ZGFwZXN0
-----END CERTIFICATE-----
  key: |
-----BEGIN RSA PRIVATE KEY-----
MIIJKQIBAAKCAgEAhdEeZDF8jIvm/GxAvOOTlpUfn+B2zPkgu+X62x6U3NMtfY+E
-----END RSA PRIVATE KEY-----
```

After configuring the TLS provisioning method in Ansible and launching the MiCADO Master Node, users should be able to access the MiCADO dashboard over HTTPS only. Users can see the certificate related information through browser displays an example of such information in the case of a self-signed certificate).

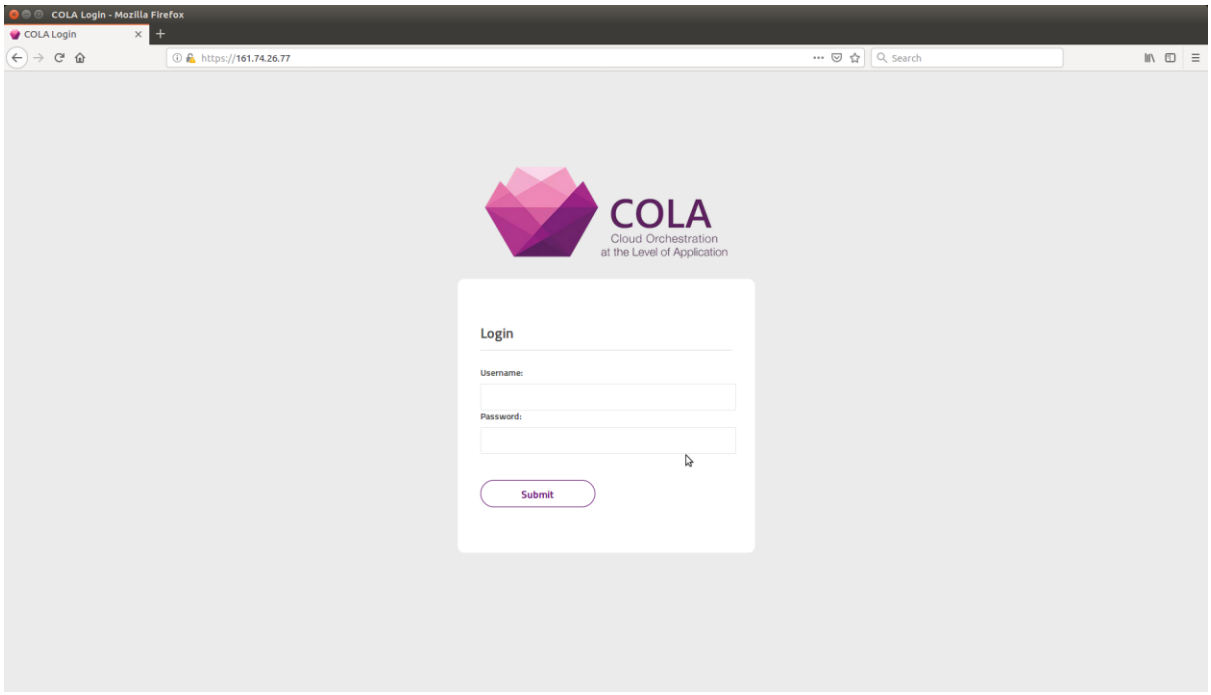

Figure 1 MiCADO dashboard login

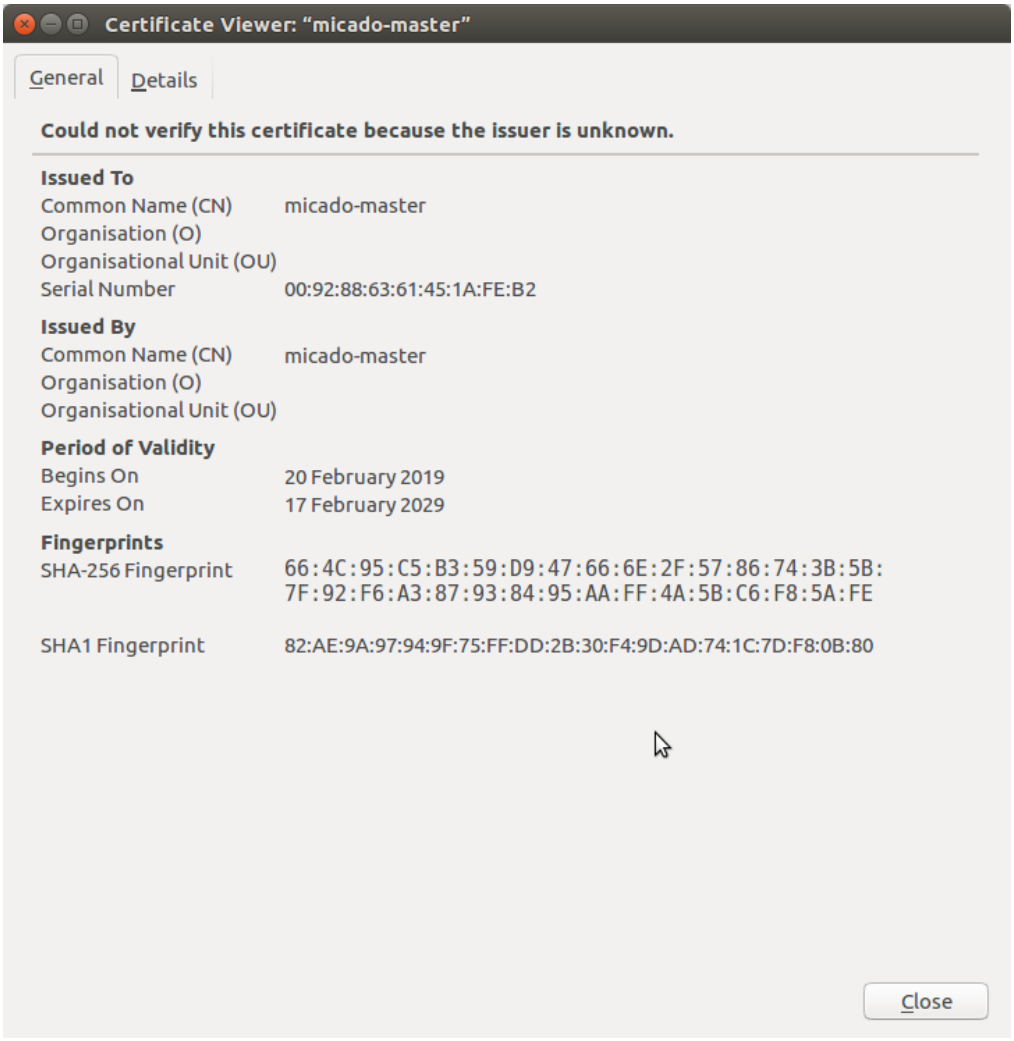

Figure 2 Certificate-related information

2.4.2 GL2 - Adding the first user with 'admin' role into MiCADO

Only users with valid usernames and passwords should be able to deploy applications onto MiCADO. In order to define users, who will then be able to submit applications, users need to be created at the time of launching the MiCADO framework. More specifically, the user should edit the Ansible file *'ansible-micado/credentials-micado.yml'* (refer to [10] for further details on deployment instruction) to include the very first user with the 'admin' role for MiCADO. The specified user account with the 'admin' role is added and stored in the Master Node after the launch of MiCADO. After this, the specified user will then be able to submit applications to MiCADO. The following snippet demonstrates the particular specification in the ansible file.

Table 24 Creating the first user with 'admin' role

'ansible-micado/credentials-micado.yml'

```
authentication:
  # username defaults to admin
    username: [username]
    # e-mail address defaults to root@micado-master
    email: root@micado-master
    # password defaults to admin
```

```
    password: [password]
```

2.4.3 GL3 - Providing valid username and password to deploy/ un-deploy application in MiCADO

To ease the process of application deployment onto MiCADO, an automatic example script is provided in MiCADO-scale GitHub [1]. However, only the user with the 'admin' role is allowed to deploy the application, therefore, this script requires the admin credentials in the '*_settings'* file as demonstrated in the below snippet so that MiCADO can verify access before deploying the corresponding application:

Table 25 Providing username and password to deploy/ un-deploy an application

'_settings'

```
SSL_USER = [username]
SSL_PASS = [password]
```

The application deployment will be successfully with the following notification, if the provided username and password are authenticated and the user has the 'admin' role:

```
$ ./1-submit-tosca-nginx.sh
Settings used:
  MICADO_MASTER: 161.74.26.222
  MICADO_PORT: 443
  APP_ID: httptest
  SSL_USER: admin
  SSL_PASS: (hidden)
Submitting nginx.yaml to MiCADO at 161.74.26.222 with appid "httptest"...
{
  "data": [],
  "message": "successfully launched app httptest",
  "status_code": 200
}
```

Figure 3 Deploy an application

The un-deployment process follows the same pattern, i.e. only a user with the 'admin' role is allowed to un-deploy an application. An un-deployment script [1] is provided to ease the process. The following message can be seen after the successful un-deployment process.

```
$ ./4-undeploy-nginx.sh
Settings used:
  MICADO_MASTER: 161.74.26.222
  MICADO_PORT: 443
  APP_ID: httptest
  SSL_USER: admin
  SSL_PASS: (hidden)
Deleting app with id "httptest" from MiCADO at 161.74.26.222...
{
  "data": [],
  "message": "successfully undeployed httptest",
  "status_code": 200
}
```


Figure 4 Undeploy an application

2.4.4 GL4 - Managing users in MiCADO

At the launch of MiCADO, the very first user with 'admin' role should be created. Later, while the framework is running, more users can be created. In order to create more users, SSH access is required to Master Node. Once SSH access is acquired, then different user management functions can be achieved using the following commands of MiCADO command line utility.

Table 26 Commands to manage users in MiCADO

#	Functionality	Command	Example
1	Create a user with default role 'user'	*micadoctl users add [username] [password]*	micadoctl users add alice 123
2	Change a user's role (role may be 'user' or 'admin')	*micadoctl users chrole [username] [new role]*	micadoctl users chrole alice admin
3	Delete a user	*micadoctl users del [username]*	micadoctl users del alice
4	List all users	*micadoctl users list*	
5	Reset a user's password	micadoctl users resetpwd [username]	micadoctl users resetpwd alice

2.4.5 GL5 - Providing valid username and password to access MiCADO Dashboard

MiCADO framework facilitates users to view the status of the system, such as tracking their application, system resource utilization level, worker nodes, etc. This can be achieved through accessing the MiCADO dashboard with valid user credentials. The dashboard can be accessed by all valid users irrespective of their role. Figure 5 displays the page of MiCADO dashboard after successfully logging in.

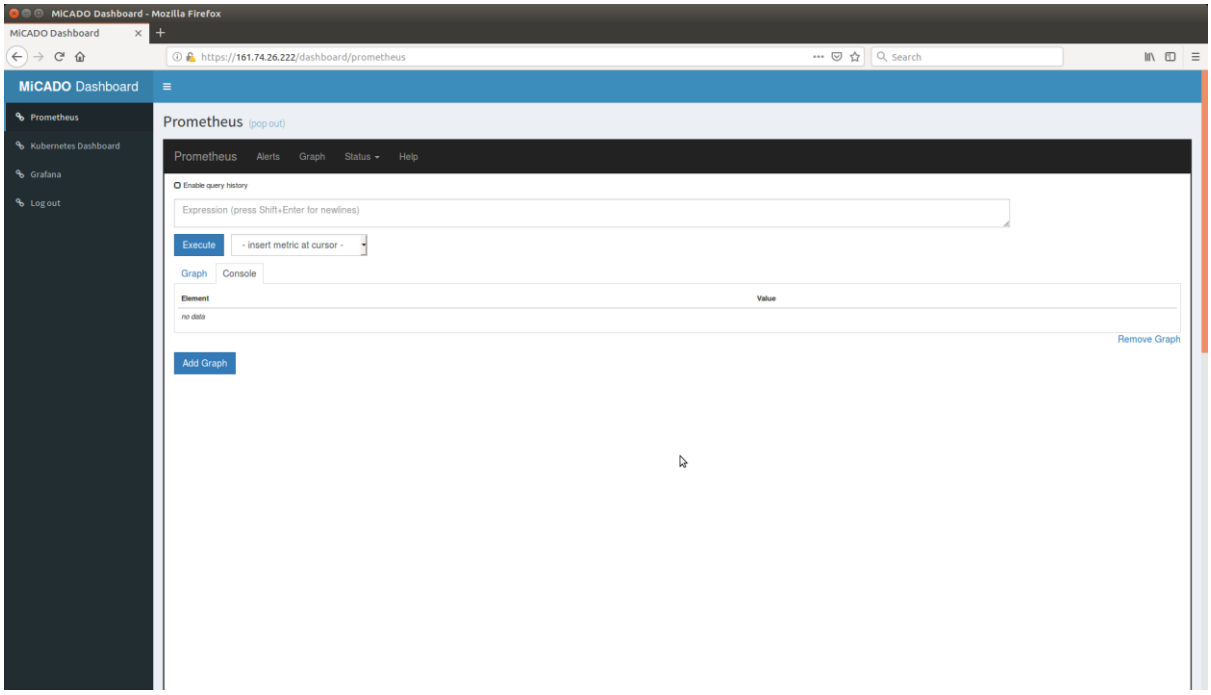

Figure 5 MiCADO dashboard

2.4.6 GL6 - Configuring L7 firewall authentication policies

L7 firewall (Zorp firewall) is installed and configured inside the master node when MiCADO is launched through an Ansible playbook. Users who launch MiCADO can customize the firewall policy by making changes to the Ansible playbook file 'ansible-micado/roles/micado-master/files/zorp/policy.py' (refer to [10] for further details on how to setup and complete a successful deployment). To demonstrate the change in a policy, let us consider an example where a user wishes to change the authentication policy. The new policy is to allow users with the role 'user' to deploy applications in contrast to what was defined previously, i.e. only users with the 'admin' role are allowed to deploy applications into the MiCADO framework. To comply with the new policy, the existing setting at the time of launching MiCADO can be changed as follows:

Table 27 Configuring authentication policies

'ansible-micado/roles/micado-master/files/zorp/policy.py'

```
class MicadoMasterHttpProxy(AuthorizingFormAuthHttpProxy):
    def config(self):
        self.auth_mapping["/toscasubmitter"] = "user" # By default, this value is "admin". Now we change it into "user"
```

Once MiCADO is launched users with the role 'user' will be able to deploy an application.

3 Security/ penetration testing

As part of the security evaluation, we have performed a detailed security examination of a sample installation of the MiCADO framework. The master node consists of a web based portal and an internet-facing REST API, and we have used multiple security credentials for the validation. This testing effort took place in May 2019. Preliminary findings were provided under a separate cover and this report presents the full results of our testing efforts and makes recommendations where appropriate.

3.1 Scope

The scope of this review was limited to the master node web application portal, the REST API and the worker nodes. This is a cloud application and the specific instantiation of the portal we were using CloudSigma as the service provider. For the sake of the test, all cloud security measures were turned off and the machine where the penetration test was performed from was placed in the same security group as the nodes of the MiCADO system.

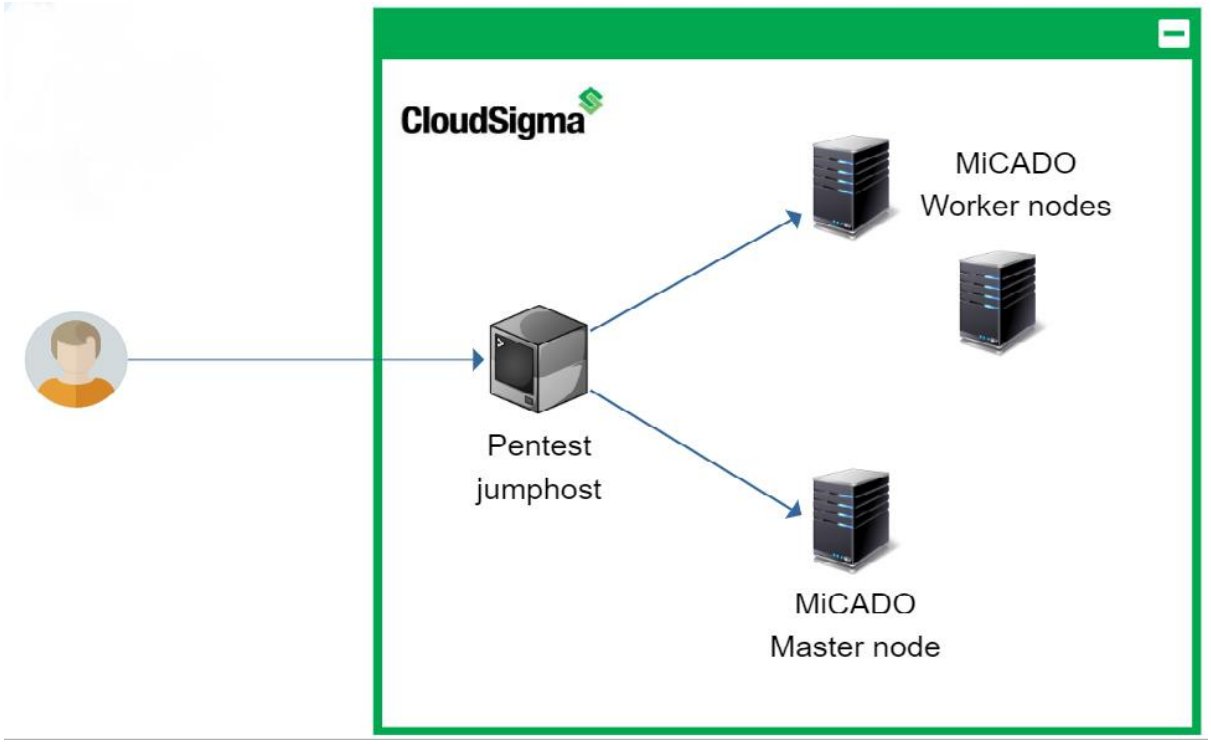

Figure 6 Penetration testing approach

The master node is Internet-facing and requires standard username and password identity elements for secure access. The landing page to the application under review was at the following addresses:

Table 28 Master node

Application	Landing page
MiCADO master node	https://178.22.68.112/

Our testing included both unauthenticated as well as authenticated testing. For the purpose of our testing we were provided with 2 unique accounts for the MiCADO Master Node. These

accounts were used to test the application's internal security controls. These accounts are displayed in the table below.

Table 29 Test Accounts

Account name	Role
admin	Administrator
user1	User

For the assessment, the *Wordpress* demo application was deployed in MiCADO.

At the time of testing, the Occopus component was not able to configure security groups for the deployed worker nodes, so an Internet-facing scan of the deployed application was not possible. For future assessments, it is recommended that the application is configured completely and all functionality is verified.

3.2 Summary of Findings

In performing a detailed application penetration study against the MiCADO application suite, **we have identified few issues of concern and overall found the application to be built around a solid security model**. Throughout this report we provide brief descriptions of each testing category and provide more detailed where our findings were negative. The below table shows a breakdown of the vulnerabilities identified based on category and severity of risk. This table is followed by a detailed breakdown outlining each category.

Table 30 Summary of Findings

Testing category	High	Medium	Low
Web Site Pilfering	-	-	-
Files Guessing attacks	-	-	-
Modifying inputs and Parameter Tampering	-	-	-
Bypassing client side validation	-	-	-
Hidden field identification and tampering	-	-	-
Cookie Abuse	-	-	-
Session Hijacking	-	-	-
URL Jumping	-	-	-
Cross Site Scripting	-	-	-
Directory browsing	-	-	-
SQL Injection	-	-	-
Functional Design Issues	-	-	-
System & Software vulnerabilities	-	-	5

3.2.1 Web Site Pelfering

Attackers often gain much information simply by what is stored in the content of the web site files that are transferred to the client's browser. We spidered the Master Node to make certain we understood the layout of the application before we started any actual attacks. We used regular expressions to search through the body of the html and javascript to identify any information that might be useful to an attacker.

We searched for many common issues including:

- Unnecessary and revealing programmer comments (none found)
- IP addresses (none found)
- Email addresses (none found)
- Raw SQL queries (none found)
- Database connection strings (none found)
- Hidden Fields (none found)

Conclusion:

We performed full text searches of crawl results looking for sensitive information within the HTML code. These tests did not reveal anything that would be of use to an attacker.

3.2.2 File Guessing attacks

It is sometimes possible to find interesting content on a web site simply by "snooping" around. Sometimes there are backup files of older versions of live code, or perhaps vulnerable sample application pages left on the web site. When accessing sensitive application data, this application relies on dynamic tokens that change with each request. This behaviour makes fuzzing for application data an impractical test case, although we did still test for common file names using tools such as Burb, DirBuster and Acunetix.

Conclusion:
We attempted various URL brute-force testing for common file names but none were successful in identifying any hidden or otherwise undisclosed files. We ran Burp, DirBuster and Acunetix scans in search of useful files, but did not succeed in identifying anything, which would aid an attacker.

3.2.3 Modifying input choices and Parameter Tampering

Web applications often pre-populate variables for users either based on the user's identity, pre-populated values in hidden fields, or as a result of user selection from a list. The assumption is that these values will be presented to the server in a controlled state; however, it is possible to intercept the client initiated GET or POST and change these values. Since there is an assumption of trust in this process, developers sometimes treat this client provided input with less scrutiny tha n input directly typed by the user. We are therefore interested in input generated on the client side of the connection, and spent time tampering with those inputs to see if we can trick the application into bypassing certain authorization controls. This attack method is commonly referred to as Parameter Tampering.

Conclusion:
The variables passed to the applications are properly validated and no vulnerabilities could be uncovered.

3.2.4 Bypassing client side validation

Validating user input is an important security control which must be performed or a web application will be susceptible to a large range of injection-based attacks and potentially authentication or authorization bypass attacks. There are two methods of validation checks that a developer can employ. The first is to check input within the client via the HTML source or scripting language source loaded into the browser. These checks will be performed when a user clicks a submit button on the page, or when focus moves from an input field. The other form of validation check is a server-side check. This type of check is performed within the receiving application on the server. These checks are typically performed after the page is submitted to the server, but before the input choices are submitted to lower level processes. The major difference between these two methods is that a malicious user can alter client-side checks but would have no control over server-side checks.

Conclusion:
The application suite seems to be solidly performing server-side validation of input parameters. Most components that submit information are calling a REST API in the background and proper input controls are implemented in the APIs. No input validation problems were uncovered during the testing.

3.2.5 Hidden field identification and tampering

Developer s sometimes hide information in hidden fields within the HTML of the web page. These hidden fields are not meant to be used as security control, and doing so constitutes attempting to provide "security through obscurity". Just because the field does no t show up on the rendered page in the browser, does not mean it cannot be seen by viewing the page source or even intercepted and tampered with.

Conclusion:

We were unable to leverage hidden fields to aid in a successful attack.

3.2.6 Cookie Abuse

Cookies are used to store static information on a per user/browser basis or on a per session basis. It is session cookies that control much of the security of modern web applications. Modification of cookies, often called poisoning, has been a common attack vector since their inception into web-based applications. Using cookie poisoning attacks, an attacker could gain unauthorized information about another user or steal a user's identity.

One notable cookie is generated for the Master Node Dashboard which handles session state, named **ZorpSession**.The ZorpSession cookie is reset upon a proper logout. We attempted various types of cookie abuses such as:

- removing the cookie post-authentication,
- swapping out cookies with those of other known and active sessions, and
- forcing out of sequence authentication submittals (bypassing logout)

Tampering with the cookies would either cause a forced re-authentication or some other error.

Conclusion:

Cookies that control session state and or authorization appear to be properly protected from tampering attacks. We were unsuccessful in finding any issues in the way that cookies were being implemented on the MiCADO Master Node dashboard.

3.2.7 Session Hijacking

Session hijacking is the act of taking control of a user session after successfully obtaining or generating an authentication session ID. Session hijacking involves an attacker using captured, brute forced or reverse-engineered session IDs to seize control of a legitimate user's Web application session while that session is still in progress. We performed an Entropy test, to test the randomness of a couple of cookies that are generated using the **ent** command line tool.

Table 31 Session hijacking summary

Entropy = 4.875000 bits per byte. Optimum compression would reduce the size of this 32 byte file by 39 percent.   Chi square distribution for 32 samples is 256.00, and randomly would exceed this value 47.06 percent of the times.   Arithmetic mean value of data bytes is 121.9375 (127.5 = random). Monte Carlo value for Pi is 3.200000000 (error 1.86 percent). Serial correlation coefficient is -0.005049 (totally uncorrelated = 0.0).
Entropy = 5.000000 bits per byte.   Optimum compression would reduce the size of this 32 byte file by 37 percent.   Chi square distribution for 32 samples is 224.00, and randomly would exceed this value 91.97 percent of the times.

Arithmetic mean value of data bytes is 118.5312 (127.5 = random). Monte Carlo value for Pi is 4.000000000 (error 27.32 percent). Serial correlation coefficient is 0.126675 (totally uncorrelated = 0.0).
Entropy = 4.687500 bits per byte. Optimum compression would reduce the size of this 32 byte file by 41 percent. Chi square distribution for 32 samples is 304.00, and randomly would exceed this value 1.91 percent of the times. Arithmetic mean value of data bytes is 149.2188 (127.5 = random). Monte Carlo value for Pi is 1.600000000 (error 49.07 percent). Serial correlation coefficient is -0.267069 (totally uncorrelated = 0.0).
Entropy = 4.875000 bits per byte. Optimum compression would reduce the size of this 32 byte file by 39 percent. Chi square distribution for 32 samples is 256.00, and randomly would exceed this value 47.06 percent of the times. Arithmetic mean value of data bytes is 135.9688 (127.5 = random). Monte Carlo value for Pi is 3.200000000 (error 1.86 percent). Serial correlation coefficient is -0.039697 (totally uncorrelated = 0.0).
Entropy = 4.750000 bits per byte.

Optimum compression would reduce the size

of this 32 byte file by 40 percent.

Chi square distribution for 32 samples is 288.00, and randomly

would exceed this value 7.61 percent of the times.

Arithmetic mean value of data bytes is 120.2188 (127.5 = random).

Monte Carlo value for Pi is 3.200000000 (error 1.86 percent).

Serial correlation coefficient is -0.015685 (totally uncorrelated = 0.0).

Conclusion:

The session cookies used for these applications are quite strong and therefore resistant to tampering.

3.2.8 URL Jumping

URL jumping is the practice of avoiding authentication and authorization simply by pointing directly to a known link. For instance, a user might be logged into a site and enter /admin at the end of the URL bar and be directed to the administrative page, even though the user is not an administrator. This sort of attack will work when security controls, such as session management, are not in play, and security is provided through obfuscation only.

Conclusion:

We did not uncover any URL jumping issues within the MiCADO management interfaces.

3.2.9 Cross Site Scripting

Cross site scripting is an attack vector that takes advantage of dynamically generated Web pages. In an XSS attack, a Web application is sent with a script that activates when it is read by an unsuspecting user's browser or by an application that has not protected itself against cross-site scripting. Because dynamic Web sites rely on user input, a malicious user can input malicious script into the page by hiding it within legitimate requests. Common exploitations include search engine boxes, online forums and public-accessed blogs. Once XSS has been launched, the attacker can change user settings, hijack accounts, poison cookies with malicious code, expose TLS connections, access restricted sites and even launch false advertisements.

Conclusion:

We ran automated tests in search of cross site scripting issues on the MiCADO Master Node, but found no evidence of this vulnerability.

3.2.10 Directory browsing

Directory browsing is an information gathering attack which leverages an administrative misconfiguration in a web server which allows listing of directory contents. This is bad practice as it provides a would-be attacker far too much information. Most web servers are configured out-of-the-box with directory browsing turned on. As a result, this vulnerability is still often found in the wild.

Conclusion:

We identified no instances where directory browsing was allowed on the MiCADO Master Node.

3.2.11 SQL Injection

SQL Injection is an attack method which allows an attacker to inject SQL code through some method of client supplied input. This injected code will be concatenated with valid code on the server side to change the SQL query to allow some form of unauthorized access, data mining, or code execution. SQL injection flaws can often be found on authentication pages and allow unauthorized access to a vulnerable site.

Conclusion:

We performed both automated and manual attacks, but did not find any SQL Injection flaws on the MiCADO dashboard web application.

3.2.12 Logical Design Issues

Logical design issues are programmatic flaws within the application, which cause the application to operate in some way other than how the programmer originally planned, and that could pose a threat to the security of the site. This category of testing is somewhat of a catch-all where any issues that do not fall neatly into one of the earlier categories can be presented.

Conclusion:

Overall the application seems very soundly built. We did not uncover any logical flaws.

3.2.13 System and software vulnerabilities

Web server software and host operating systems where web applications exist are sometimes misconfigured in ways that impose risk on the web application. We use various vulnerability scanning methods to test for issues within the application's system components.

Table 32 TCP Timestamping

MiCADO components have tcp timestamping (RFC1323) enabled	
Risk	Low
Complexity	High
Summary	The remote host implements TCP timestamps and therefore allows to compute the uptime. This is an information disclosure type of error and cannot be exploited on its own.
Mitigation	To disable TCP timestamps on L inux add the line 'net.ipv4.tcp_timestamps = 0' to /etc/sysctl.conf. Execute 'sysctl -p' to apply the settings at runtime.

Table 33 X-Frame-Options header

The MiCADO Dashboard doesn't use the X-Frame-Options header	
Risk	Low
Complexity	High
Summary	The X-Frame-Options HTTP response header can be used to indicate whether or not a browser should be allowed to render a page in a <frame>, <iframe>, <embed> or <object> . Sites can use this to avoid clickjacking attacks, by ensuring that their content is not embedded into other sites.
Mitigation	Set the `X-Frame-Options: sameorigin` HTTP header on the dashboard to make sure that page can only be displayed in a frame on the same origin as the page itself.

Table 34 X-XSS-Protection header

The MiCADO Dashboard doesn't use the X-XSS-Protection header	
Risk	Low
Complexity	High
Summary	The HTTP X-XSS-Protection response header is a feature of Internet Explorer, Chrome and Safari that stops pages from loading when they detect reflected cross-site scripting (XSS) attacks. Although these protections are largely unnecessary in modern browsers when sites implement a strong Content-Security-Policy that disables the use of inline JavaScript ('unsafe-inline'), they can still provide protections for users of older web browsers that don't yet support CSP.
Mitigation	Set the `X-XSS-Protection: 1; mode=block`

	HTTP header on the dashboard to make sure that if the browser detects a possible XSS attack , the page is blocked without further sanitization attempts.

Table 35 Strict-Transport-Security header

The MiCADO Dashboard doesn't use the Strict-Transport-Security header	
Risk	Low
Complexity	High
Summary	HTTP Strict Transport Security lets a web site inform the browser that it should never load the site using HTTP and should automatically convert all attempts to access the site using HTTP to HTTPS requests instead. As the MiCADO Master redirects HTTP requests to HTTPS automatically it would spare system resources and ensure the integrity of all client communication to use this feature.
Mitigation	Set the `Strict-Transport-Security: max-age=31536000; includeSubDomains` HTTP header on the dashboard to make sure that all clients use HTTPS exclusively to access it.

Automatic testing revealed another potentially missing HTTP header, that would increase overall security, but due to the number of 3rd party tools used in MiCADO it is not worth the effort to implement the X-Content-Type-Options header.

Conclusion:

No actual vulnerabilities were discovered during the testing, but the above areas of concern should be addressed in future releases.

4 Security Performance Evaluation

This section briefly summarizes the performance related evaluation of the MiCADO framework in the context of security. Hence, the conducted evaluation tests for this purpose measure the overhead caused by the various security enablers in the MiCADO framework. All these tests were performed on CloudSigma environment with the latest version of the MiCADO framework (v 0.7.3). The description and results of each test is as follow:

1. Full build deployment: This test was conducted to estimate the time required to deploy the MiCADO framework with and without the security enablers. In this case, the deployment was performed using the full build option, i.e. when all components of MiCADO framework are directly built from code. The obtained results are the following:

Table 36 Full build deployment times

Test case	Result
Without security enablers	6 minutes, 0 seconds
With security enablers	9 minutes, 50 seconds

2. Deployment using prepared VM image: this test is similar to the previous scenario. However, in this case the deployment of MiCADO framework is performed using the already available prepared virtual machine images, with the following results :

Table 37 Prepared VM deployment times

Test case	Result
Without security enablers	1 minute, 10 seconds
With security enablers	2 minutes, 55 seconds

3. Disk size after deployment: This test was conducted to measure the consume disk space of MiCADO framework with and without security enablers. The obtained results are the following:

Table 38 Deployment size on disk

Test case	Result
Without security enablers	2.7 GB
With security enablers	4.7 GB

4. Operational: This test was conducted to examine the operational behaviour of the MiCADO framework in the presence and absence of security features of authentication and encryption handled through Zorp component. Hence, the test aimed to assess whether the Zorp based authentication and encryption will result in a severe performance overhead. The following experiment was designed to measure the performance of the system with and without authentication and encryption. In this experiment, 100 users with a Hatch rate of 20 users/sec was producing a /GET request to the MiCADO dashboard. The performance of the system against this experiment are recorded for the following two cases: (1) with Zorp: where the incoming connections are handled through Zorp container, which wraps the

connection in TLS, perform authentication and forward the requests to the dashboard container for processing; and (2) without Zorp: where the requests were forwarded directly to the dashboard container without any authentication and encryption. The results for both scenarios are summarized in Figure 6.

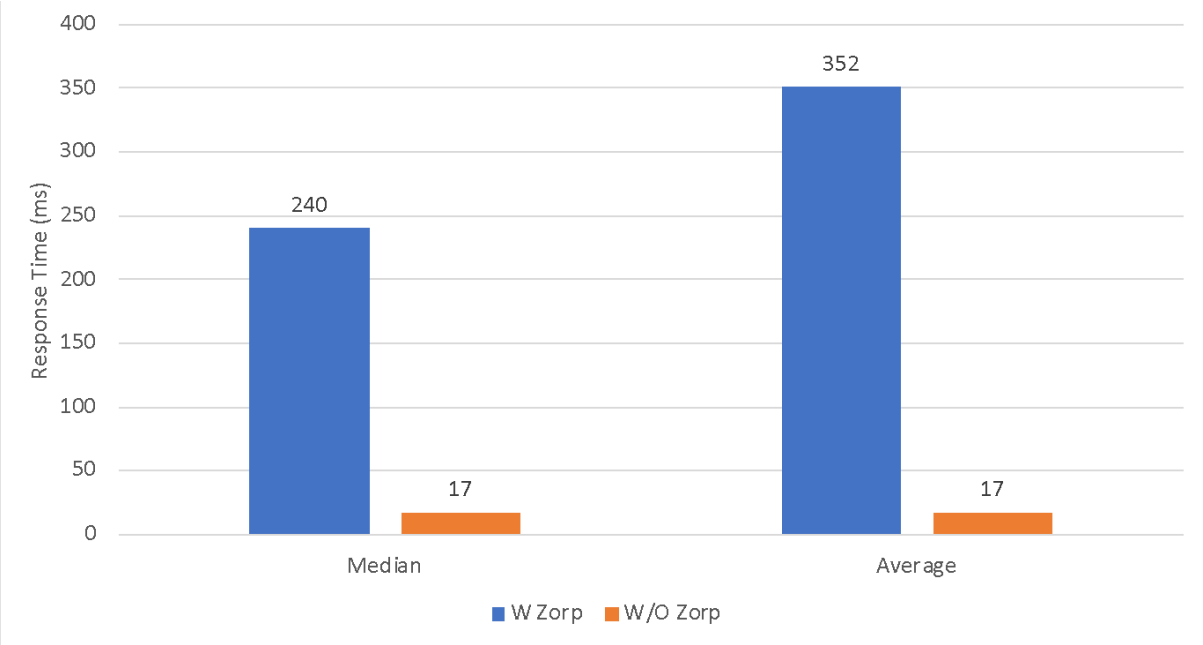

Figure 7 Summarised result of MiCADO security evaluation performance tes

The scope of this deliverable was to provide a snapshot of the security of the MiCADO framework. We reviewed the security enablers developed and integrated within WP7 of the COLA project, in terms of:

- Security Requirements collected from the COLA use case partners;
- NIST 800-53 compliance control reference
- Security configuration guidelines
- Resilience to known vulnerabilities, though enabler penetration testing.
- Performance impact of the integrated security enablers.

The evaluation has shown that the enablers address a large part of the security requirements collected from the use-case partners. The COLA modular security architecture allows it to selectively enable or disable the use of a subset of the security enablers. This allows to balance the security and maintainability of the individual MiCADO deployments. Likewise, the modular security architecture allows the development of additional security enablers tailored to the individual deployment requirements.

Penetration testing was the second major technical aspect of the security evaluation of the enablers developed within WP7. Penetration testing revealed no vulnerabilities among the analysed enablers. Moreover, the penetration testing results have highlighted several improvement areas to be addressed in the future releases of MiCADO.

Digital systems security is commonly viewed as a *cost*, reflected through higher implementation and software validation efforts during the development phase, as well as a performance overhead during the operation of the software. Offsetting this cost by using security enablers to improve performance is untrivial and sometimes impossible. We did not

evaluate *the implementation and software validation costs* to develop security enablers for the MiCADO framework. To measure the *operational performance* costs of the security enablers for the MiCADO framework, we performed a security performance evaluation. The results of this evaluation show an increase in the disk size of the code base, as well as longer processing times for incoming connections in two different scenarios. These results will help MiCADO administrators decide the performance-security trade-off considering the specific deployment contexts.

5 References

[1] Automatic example scripts to deploy applications in MiCADO, https://github.com/micado-scale/ansible-micado/tree/master/testing/nginx

[2] Zorp firewall policy configuration for MiCADO master node, https://github.com/micado-scale/ansible-micado/blob/cd0f02ed716114f2a49281fc649100b645f00f0d/roles/micado-master/files/zorp/policy.py

[3] Deliverable 7.1 - COLA security requirements

[4] Deliverable 7.2 - MiCADO security architecture specification

[5] Deliverable 7.3 - Design of application level security classification formats and principles

[6] Deliverable 7.4 - Security policy formats specification

[7] Deliverable 7.5 – MiCADO security modules reference implementations

[8] NIST 800-53: Security and Privacy Controls for Information Systems and Organisations, Available from: https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/draft

[9] Application Description Templates (ADT), https://github.com/micado-scale/tosca

[10] MiCADO Deployment guideline, Available from: https://micado-scale.readthedocs.io/en/latest/deployment.html

[11] Flask-User, https://flask-user.readthedocs.io/en/latest/

[12] MiCADO Github repository, Available from: https://github.com/micado-scale

5. Appendixes

A) Appendix A – Nikto scan report

178.22.68.112 / 178.22.68.112 port 443

Target IP	178.22.68.112
Target hostname	178.22.68.112
Target Port	443
HTTP Server	
Site Link (Name)	https://178.22.68.112:443/
Site Link (IP)	https://178.22.68.112:443/
URI	/
HTTP Method	GET
Description	The anti-clickjacking X-Frame-Options header is not present.
Test Links	https://178.22.68.112:443/ https://178.22.68.112:443/
OSVDB Entries	OSVDB-0

URI	/
HTTP Method	GET
Description	The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
Test Links	https://178.22.68.112:443/ https://178.22.68.112:443/
OSVDB Entries	OSVDB-0

URI	/
HTTP Method	GET
Description	The site uses SSL and the Strict-Transport-Security HTTP header is not defined.
Test Links	https://178.22.68.112:443/ https://178.22.68.112:443/
OSVDB Entries	OSVDB-0

URI	/
HTTP Method	GET
Description	The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type

Test Links	https://178.22.68.112:443/ https://178.22.68.112:443/
OSVDB Entries	OSVDB-0

URI	/clientaccesspolicy.xml
HTTP Method	GET
Description	lines
Test Links	https://178.22.68.112:443/clientaccesspolicy.xml https://178.22.68.112:443/clientaccesspolicy.xml
OSVDB Entries	OSVDB-0

Host Summary

Start Time	2019-06-04 12:13:41
End Time	2019-06-04 12:15:49
Elapsed Time	128 seconds
Statistics	446 requests, 20 errors, 5 findings

Scan Summary

Software Details	Nikto 2.1.6
CLI Options	-host https://178.22.68.112 -Display V -F html -output niktoscan.html -Tuning 0 1 2 3 4 5 7 8 a b c
Hosts Tested	1
Start Time	Tue Jun 4 12:13:41 2019
End Time	Tue Jun 4 12:15:49 2019
Elapsed Time	128 seconds

B) Appendix B – OpenVAS test report for the MiCADO Master Node

Summary

This document reports on the results of an automatic security scan. The report first summarises the results found. Then, for each host, the report describes every issue found. Please consider the advice given in each description, in order to rectify the issue.

Vendor security updates are not trusted.

Overrides are on. When a result has an override, this report uses the threat of the override.

Information on overrides is included in the report.

Notes are included in the report.

This report might not show details of all issues that were found. It only lists hosts that produced issues. Issues with the threat level "Log" are not shown. Issues with the threat level "Debug" are not shown. Issues with the threat level "False Positive" are not shown. Only results with a minimum QoD of 70 are shown.

This report contains result 1 of the 1 results selected by the filtering above. Before filtering there were 33 results.

All dates are displayed using the timezone "Coordinated Universal Time", which is abbreviated "UTC".

Scan started: **Thu May 30 15:15:53 2019 UTC**

Scan ended: Thu May 30 15:45:17 2019 UTC

Task: Full test on Micado master node

Host Summary

Host	Start	End	High	Medium	Low	Log	False Positive
178.22.70.163 (host-163-70-22-178.cloudsigma.net)	May 30, 15:16:20	May 30, 15:45:17	0	0	1	0	0
Total: 1			0	0	1	0	0

Results per Host

Host 178.22.70.163

Scanning of this host started at: Thu May 30 15:16:20 2019 UTC

Number of results: 1

Port Summary for Host 178.22.70.163

Service (Port)	Threat Level
general/tcp	Low

Security Issues for Host 178.22.70.163

general/tcp

Low (CVSS: 2.6)

NVT: TCP timestamps (OID: 1.3.6.1.4.1.25623.1.0.80091)

Summary

The remote host implements TCP timestamps and therefore allows to compute the uptime.

Vulnerability Detection Result

It was detected that the host implements RFC1323.

The following timestamps were retrieved with a delay of 1 seconds in-between:

Packet 1: 23390338

Packet 2: 23390712

Impact

A side effect of this feature is that the uptime of the remote host can sometimes be computed.

Solution

Solution type: Mitigation

To disable TCP timestamps on linux add the line 'net.ipv4.tcp_timestamps = 0' to /etc/sysctl.conf. Execute 'sysctl -p' to apply the settings at runtime.

To disable TCP timestamps on Windows execute 'netsh int tcp set global timestamps=disabled'

Starting with Windows Server 2008 and Vista, the timestamp can not be completely disabled.

The default behavior of the TCP/IP stack on this Systems is to not use the Timestamp options when initiating TCP connections, but use them if the TCP peer that is initiating communication includes them in their synchronize (SYN) segment.

See the references for more information.

Affected Software/OS

TCP/IPv4 implementations that implement RFC1323.

Vulnerability Insight

The remote host implements TCP timestamps, as defined by RFC1323.

Vulnerability Detection Method

Special IP packets are forged and sent with a little delay in between to the target IP. The responses are searched for a timestamps. If found, the timestamps are reported.

Details: TCP timestamps (OID: 1.3.6.1.4.1.25623.1.0.80091)

Version used: $Revision: 14310 $

References

Other: http://www.ietf.org/rfc/rfc1323.txt

http://www.microsoft.com/en-us/download/details.aspx?id=9152

C) Appendix C – OpenVAS test report for the MiCADO Worker Node

Summary

This document reports on the results of an automatic security scan. The report first summarises the results found. Then, for each host, the report describes every issue found. Please consider the advice given in each description, in order to rectify the issue.

Vendor security updates are not trusted.

Overrides are on. When a result has an override, this report uses the threat of the override.

Information on overrides is included in the report.

Notes are included in the report.

This report might not show details of all issues that were found. It only lists hosts that produced issues. Issues with the threat level "Log" are not shown. Issues with the threat level "Debug" are not shown. Issues with the threat level "False Positive" are not shown. Only results with a minimum QoD of 70 are shown.

This report contains result 1 of the 1 results selected by the filtering above. Before filtering there were 29 results.

All dates are displayed using the timezone "Coordinated Universal Time", which is abbreviated "UTC".

Scan started: **Thu May 30 15:15:19 2019 UTC**

Scan ended: Thu May 30 15:39:31 2019 UTC

Task: Full test on Micado worker node

Host Summary

Host	Start	End	High	Medium	Low	Log	False Positive
31.171.246.220 (host-220-246-171-31.cloudsigma.net)	May 30, 15:15:48	May 30, 15:39:31	0	0	1	0	0
Total: 1			0	0	1	0	0

Results per Host

Host 31.171.246.220

Scanning of this host started at: Thu May 30 15:15:48 2019 UTC

Number of results: 1

Port Summary for Host 31.171.246.220

Service (Port)	Threat Level

general/tcp Low

Security Issues for Host 31.171.246.220

general/tcp

Low (CVSS: 2.6)

NVT: TCP timestamps (OID: 1.3.6.1.4.1.25623.1.0.80091)

Summary

The remote host implements TCP timestamps and therefore allows to compute the uptime.

Vulnerability Detection Result

It was detected that the host implements RFC1323.

The following timestamps were retrieved with a delay of 1 seconds in-between:

Packet 1: 531176

Packet 2: 531497

Impact

A side effect of this feature is that the uptime of the remote host can sometimes be computed.

Solution

Solution type: Mitigation

To disable TCP timestamps on linux add the line 'net.ipv4.tcp_timestamps = 0' to /etc/sysctl.conf. Execute 'sysctl -p' to apply the settings at runtime.

To disable TCP timestamps on Windows execute 'netsh int tcp set global timestamps=disabled'

Starting with Windows Server 2008 and Vista, the timestamp can not be completely disabled.

The default behavior of the TCP/IP stack on this Systems is to not use the Timestamp options when initiating TCP connections, but use them if the TCP peer that is initiating communication includes them in their synchronize (SYN) segment.

See the references for more information.

Affected Software/OS

TCP/IPv4 implementations that implement RFC1323.

Vulnerability Insight

The remote host implements TCP timestamps, as defined by RFC1323.

Vulnerability Detection Method

Special IP packets are forged and sent with a little delay in between to the target IP. The responses are searched for a timestamps. If found, the timestamps are reported.

Details: TCP timestamps (OID: 1.3.6.1.4.1.25623.1.0.80091)

Version used: $Revision: 14310 $

References

Other: http://www.ietf.org/rfc/rfc1323.txt

http://www.microsoft.com/en-us/download/details.aspx?id=9152

D) Appendix D – Wapiti test report for the MiCADO Master Node with Domain Scope

Wapiti vulnerability report

Target: https://178.22.68.112/

Date of the scan: Tue, 04 Jun 2019 09:38:48 +0000. Scope of the scan: domain

Summary

Category	Number of vulnerabilities found
SQL Injection	0
Blind SQL Injection	0
File Handling	0
Cross Site Scripting	0
CRLF Injection	0
Commands execution	0
Htaccess Bypass	0
Backup file	0
Potentially dangerous file	0
Server Side Request Forgery	0
Internal Server Error	0
Resource consumption	0

Wapiti 3.0.1 © Nicolas SURRIBAS 2006-2018

E) Appendix E – Wapiti test report for the MiCADO Master Node with Folder Scope

Wapiti vulnerability report

Target: https://178.22.68.112/

Date of the scan: Tue, 04 Jun 2019 09:39:24 +0000. Scope of the scan: folder

Summary

Category	Number of vulnerabilities found
SQL Injection	0
Blind SQL Injection	0
File Handling	0
Cross Site Scripting	0
CRLF Injection	0
Commands execution	0
Htaccess Bypass	0
Backup file	0
Potentially dangerous file	0
Server Side Request Forgery	0
Internal Server Error	0
Resource consumption	0

Wapiti 3.0.1 © Nicolas SURRIBAS 2006-2018

F) Appendix F – Wapiti test report for the MiCADO Master Node with Page Scope

Wapiti vulnerability report

Target: https://178.22.68.112/

Date of the scan: Tue, 04 Jun 2019 09:38:25 +0000. Scope of the scan: page

Summary

Category	Number of vulnerabilities found
SQL Injection	0
Blind SQL Injection	0
File Handling	0
Cross Site Scripting	0
CRLF Injection	0
Commands execution	0
Htaccess Bypass	0
Backup file	0
Potentially dangerous file	0
Server Side Request Forgery	0
Internal Server Error	0
Resource consumption	0