ADUCID Server Kit Administration Guide

Version 3.0.4

Release date February 1, 2016

Antala Staška 2027/79, 140 00 Prague 4 | T +420 271 100 100 | F +420 271 100 101
Vídeňská 125, 619 00 Brno | T +420 547 100 100 | F +420 547 100 101
Sady 5. května 59, 301 00 Pilsen | T +420 271 100 100 | F +420 271 100 101
Teslova 30, 821 02 Bratislava | T +421 (2) 4821 3111 | F +421 (2) 4821 3199

ISO 9001 ISO 14001 ISO/IEC 27001 ISO/IEC 20000
CZ > Corp. ID 25313029 | Registration Court RC Brno, File no. B2113
SK > Corp. ID 35787546 | Commercial Register DC Bratislava, Section Sa file 2431/B

Table of Contents

1. Introduction

This document provides steps for the configuration and maintenance of the ADUCID Server Kit appliance. The reader of this document is assumed to have administrator-level knowledge of the Linux operating system.

2. Operating system

ADUCID Server Kit is based on the CentOS distribution (version 6) of the Linux operating system. Documentation for the operating system is available on the web (http://www.centos.org/).

3. ADUCID components

In addition to the standard components of the CentOS operating system, ADUCID Server Kit also includes following components:

- AIM
- UIM
- AIM-proxy
- Oracle Java 1.6.0

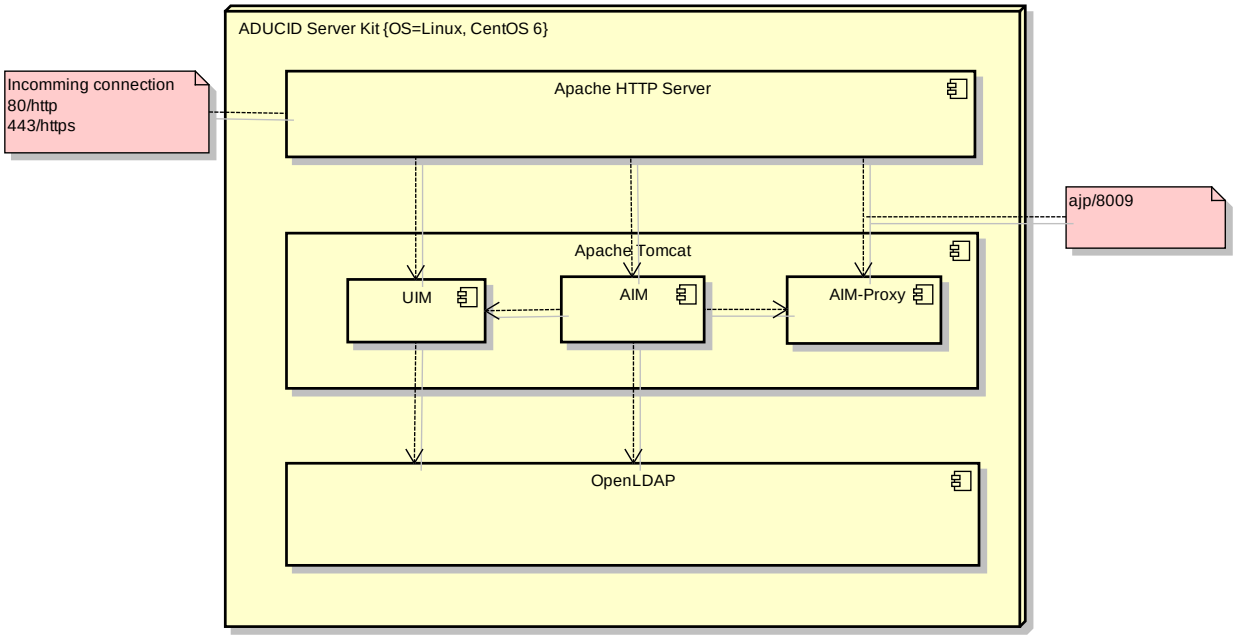

Figure 3-1 ADUCID Server Kit components

4. AIM

AIM (ADUCID Identity Machine) is the core of the server part of the ADUCID system. It runs within an application server (Tomcat) container. It communicates with its surroundings using standard web services (SOAP R3 and R4 interface).

The AIM and AIM-Proxy configuration is stored in the ADUCID.properties file (/etc/tomcat6).

Variable name	Description
AIM_host	Server URL against which the authentication is performed. The URL must include a domain name that is visible from the internet.
SPID	A unique provider identifier
ICON	A path to an icon displayed on PEIG®
IMAGE_DIR	A path to the icons shown during replication
DISPLAY_NAME	Service provider's name definition visible on PEIG
AUTH_LIFE_TIME	Maximum duration of authentication session in seconds
AUTHID_LENGTH	Length of generated authentication identifier in bytes
IMPLICIT_ACTIVITY	Implicit activity (if not specified via the R4 interface)
START_TIMEOUT	Maximum duration of the authentication start (in seconds)
PROCESS_TIMEOUT	Maximum duration of the authentication itself (in seconds)
BINDING_TIMEOUT	Maximum duration of the binding (in seconds)
BINDING_CONTROL_MODE	Enforced binding scenario code, for example ABCD, range 0-12 See [5]
HOME_URL	Target application URL to be used directly from PEIG for Android, for example http://android.aducid.eu/UIM
MEETING_ROOM_ENTER_TIME_LOCK	Maximum time that a meeting room is open for the second PEIG, default 150 (in seconds)
MEETING_ROOM_CONFIRM_TIME_LOCK	Maximum time that the second PEIG can be confirmed by the first PEIG, default 150 (in seconds)
MEETING_ROOM_MAX_REPLICA_COUNT	Maximum replica count, default 10
MEETING_ROOM_MIN_PARALLEL_LF_SUCCESS	Minimum count of success personal factor verifications on the other PEIG in time of unsuccessful personal factor verification on the verified PEIG, default 1
MEETING_ROOM_MAX_PARALLEL_LF_FAILURE	Maximum tolerated count of unsuccessful personal factor verifications on the verified PEIG in time of successful personal factor verification on the other PEIG, default 3
MEETING_ROOM_MAX_FAILURE_COUNT	Maximum tolerated total count unsuccessful personal factor verifications on the verified PEIG, default 12
LF_LOCK_DELAY	Basic personal factor lock delay, default 3600 (in seconds)
LF_LOCK_BUNDLE_COUNT	Attempts count in one batch, default 3
LF_BUFFER_TIME	Personal factor cache time, default 180 (in seconds)
LF_BUFFER_REFRESH_ENABLED	Personal factor buffer refresh flag, default true
PAYMENT_ASYMETRIC_KEY_ALGORITHM	Payment object asymmetric key algorithm, default RSA

PAYMENT_ASYMETRIC_KEY_SIZE	Payment object asymmetric key size, default 1024 (in bits)
PAYMENT_SIGNATURE_ALGORITHM	Payment object signature algorithm, default SHA1withRSA
PAYMENT_HASH_ALGORITHM	Payment object hash algorithm, default SHA-1
ldap.url	URL of the LDAP database
ldap.username	Username
ldap.password	Password
ldap.baseDN	baseDN
ldap.rightConfigDN	Location of user attribute sets definition
ldap.securityProfileDN	Security profile locations
ldap.identityDN	Identity locations
ldap.userDN	User locations
ldap.userProxyDN	Location of binding objects between identity and user
ldap.ilidDN	ilid identifier locations
ldap.roleDN	Role locations (optional)
ldap.applicationDN	Application locations (optional)
ldap.permissionDN	permissionDN location (optional)
ldap.ILSecurityProfileDN	ilid security profile locations
ldap.legacyProfileDN	Legacy security profile locations
ldap.userAttrSpecDN	UIM application layout definition location (optional)
db.path	Path to the HyperSQL database, where AIM events are persisted, default file:/opt/aducid/db
db.name	Database name, default aducid
db.username	Database username, default SA. Password to the database equals to ldap.password attribute value.

Table 4-1 AIM and AIM-proxy configuration parameters

5. UIM

Configuration of the UIM application (User Identity Management) is stored in the UIM.properties file (/etc/tomcat6).

Variable name	**Description**
referral	An attribute of the AducidUser object used to establish a relationship between identity and user
defaultKeyValidityHours	Maximum expiration period for the activation key (in hours)
AIM_proxy_url	URL pointing to a location where the AIM-Proxy component is available
activationKeyLength	Activation key length (in bytes)
enableLBSecurityProfiles	Enables/disables LB security profiles support.
ilid.x.name	(Optional) Secondary service provider identifier used during registration via a link identity. The x value is provider's relative number between 1 and 99. (e.g. ilid.1.name=AIM-SP1). Registration using a link identity is disabled by default.

Variable name	Description
ilid.x.url	(Optional) Secondary service provider URL used during registration via a link identity. The x value is provider's relative number between 1 and 99. (e.g. ilid.1.url=http://www.aducid.eu:80). The format of the URL is http(s)://hostname:port. In order for the registration via link identity to be enabled in UIM, both values of the pair (name and URL) must be specified.
commonNameTemplate	(Optional) Template allowing to use own mechanism for composing CN. For example "{sn}, {givenName}" will produce "Smith, John". The default is "{givenName} {sn}".

Table 5-1 UIM configuration parameters

With regards to user attributes, the UIM application is fully configurable (including the appearance, role permissions and attribute set). The configuration is located in the settings.xml file of the web application and has the following structure. If a new application needs to be added for which it is necessary to manage user attributes, the file needs to be reconfigured.

This file is located in `/var/lib/tomcat6/webapps/UIM/WEB-INF/classes/` folder. Because this folder is not persistent and is deleted during upgrade, we strongly recommend to create backup folder `/etc/tomcat6/customization`. Then create link to destination file in this folder. Doing so, you will have backup file, which is not replaced by upgrade. Here is an example how to do it:

```
mkdir /etc/tomcat6/customization
cd /etc/tomcat6/customization
ln /var/lib/tomcat6/webapps/UIM/WEB-INF/classes/settings.xml
```

It is not possible to use symlinks in this case, because tomcat is checking if settings.xml is regular file.

Let's start with an example. Individual parts of the settings.xml file will be described in detail in further chapters.

5.1. Example: Adding a new application and an administrator role

The red text in the following example indicates the parts of code used when adding a new application and defining its permissions. In this example, we will add a 'PASSIVE' application and create a new role (UIM_PASSIVE_AAM) that will have the rights to modify user accounts for the PASSIVE application. The role manager will then be able to assign the new role to a particular person in the UIM.

We want the application visible within UIM under localized 'PASSIVE' text and with the following attributes:

Attribute name	Alias	LDAP attribute	Write permission
Given Name	givenName	givenName	R
Surname	sn	sn	R
E-mail	mail	mail	R/W
Cell phone	mobile	mobile	R/W
Active	active	destinationIndicator	R/W
Organization	organization	o	R/W
Person Identifier	aducidGUID	aducidGUID	W

The configuration then looks as follows.

```
<?xml version="1.0" encoding="UTF-8"?>
<settings>
  <applications>
    <application id="UIM" cs="UIM" en="UIM">
    ...
    </application>
    <application id="PASSIVE" cs="PASSIVE" en="PASSIVE">
        <layout>
            <attribute alias="givenName" name="givenName" cs="Jméno"
                en="First name" validators="required" type="INPUT" />
            <attribute alias="sn" name="sn" cs="Příjmení" en="Surname"
                validators="required" type="INPUT" />
            <attribute alias="mail" name="mail" cs="E-mail" en="E-mail"
                validators="required email" type="INPUT" />
            <attribute alias="mobile" name="mobile"
                cs="Mobil" en="Cell phone" validators="required phoneCZ"
                type="INPUT" />
            <attribute alias="active" name="destinationIndicator"
                cs="Aktivní" en="Active"
                validators="" type="CHECKBOX" />
            <attribute alias="organization" name="o"
                cs="Organizace" en="Organization"
                validators="required" type="SELECT" multiple="false"
                codelist="ORGANIZATION" />
            <attribute alias="aducidGUID" name="aducidGUID" type="HIDDEN" />
        </layout>
        <codelists>
            <codelist id="ORGANIZATION">
                <item value="O1" cs="Organizace 1" en="Organization 1" />
                <item value="O2" cs="Organizace 2" en="Organization 2" />
                <item value="O3" cs="Organizace 3" en="Organization 3" />
                <item value="O4" cs="Organizace 4" en="Organization 4" />
            </codelist>
        </codelists>
    </application>
  </applications>
  <roles>
    <role name="UIM_RM" application="UIM" cs="Správce rolí (UIM)" en="Role Manager (UIM)">
        <permissions>
            ...
            <!– přidání role do seznamu rolí přiřaditelných Role managerem-->
            <role name="UIM_PASSIVE_AAM" permissions="ASSIGN" />
        </permissions>
    </role>
    <role name="UIM_UM" application="UIM" cs="Správce uživatelů (UIM)"
        en="User Manager (UIM)">
        ...
    </role>
    <role name="UIM_SM" application="UIM" cs="Bezpečnostní správce (UIM)"
        en="Security Manager (UIM)">
        ...
    </role>
    <role name="UIM_U" application="UIM" cs="Uživatel (UIM)" en="User (UIM)">
        ...
    </role>
    <role name="UIM_PASSIVE_AAM" application="PASSIVE"
        cs="Správce aplikace PASSIVE" en="Application manager (PASSIVE)">
        <permissions>
            <application name="PASSIVE" permissions="MANAGE" />
            <action name="MANAGE_USERS" />
            <attribute alias="givenName" permissions="READ" />
            <attribute alias="sn" permissions="READ" />
            <attribute alias="mail" permissions="READ WRITE" />
            <attribute alias="organization" permissions="READ WRITE" />
            <attribute alias="active" permissions="READ WRITE" />
```

```
            <attribute alias="aducidGUID" permissions="READ WRITE" />
        </permissions>
    </role>
  </roles>
</settings>
```

5.2. Applications section

In the applications section, other applications visible in the UIM are defined. You can define a layout for each application that consists of attributes, their descriptions, types and potential validators. The order in the XML file defines the order in the GUI.

The following XML attributes of the ‘attribute’ element can be defined:

Attribute name	Description
Alias	Alias under which the attribute will be available to the application
name	Actual name of the attribute in LDAP
cs	Description in Czech
en	Description in English
validators	Space-delimited list of validator names. Available validators are defined in ESAPI.properties.
type	‘Html’ type of the attribute (available values: INPUT, CHECKBOX, SELECT, HIDDEN)
multiple	(Optional) This attribute is a supplemental attribute for the SELECT type that specifies whether the element is multiselect or not.
codelist	(Optional) ID of a dial that is used to display the SELECT box.

The ‘codelists’ element contains the definitions of individual dial values for the SELECT type attribute.

The following XML attributes of the ‘codelist’ and ‘item’ elements can be defined:

Attribute name	Description
id	Codelist: Dial identifier for the relationship at the SELECT type attribute
value	Item: Dial value
cs	Item: Description in Czech
en	Item: Description in English

5.3. Roles section

The ‘roles’ section contains role definitions, individual attribute visibility and permissions for actions in the UIM (‘permissions’ section).

When establishing a new application within the UIM, at least one role must be assigned to it for managing the application. For this, the ‘roles’ section is available. It lets you define all roles for the UIM application.

The following XML attributes of the ‘role’ element can be defined:

Attribute name	Description
name	Name of the assigned role selected from roles enabled for administration.

application	The name of the specified application to which the role is assigned.
cs	Description in Czech
en	Description in English

The following set of permissions can be defined for each role:

5.3.1. Application administration permissions

Application element

Attribute name	Description
name	The name of the application you want to manage.
permissions	MANAGE constant

5.3.2. Permission to access individual UIM sections

Action element

Attribute name	Description
name	MANAGE_USERS constant

5.3.3. Permission to access selected user attributes

Attribute element

Attribute name	Description
Alias	The alias defined in the applications section.
Permissions	A set of space-delimited permissions for the attribute (READ WRITE constants)

5.3.4. Role assignment permission

When adding a new role, it is necessary to incorporate it into a set of roles available for the Role Manager to assign in the UIM.

Role element

Attribute name	Description
Name	The name of the role you want to assign.
Permissions	ASSIGN constant

6. Implementing configuration for test applications

Applications that are to integrate the ADUCID technology usually require a username and a list of roles. Because such configuration is currently performed manually, two predefined files are included which implement the test application (TEST) into the AIM and UIM.

The files are located on the installation DVD in the integration\adapter-tomcat folder. The files contain a predefined configuration for UIM (described in Chapter 5) and a configuration for AIM (described in Chapter 13).

The installation consists of two steps.

- Implementing the new attribute set into AIM (AIM preparation)

- Extending UIM GUI to support test application administration

Technical steps of the installation are described in the README.txt file.

6.1. AIM preparation

With regards to AIM, this step represents a creation of an attribute set with the TEST identifier that makes the following attributes available:

- aducidGUID
- givenName
- sn
- testUsername
- testRole

The attribute set is stored in ldif format in the test-app.ldif file and is implemented directly into the LDAP ADUCID Server Kit server.

6.2. UIM preparation

This step is mandatory if you want to enter user attribute data via user administration in UIM. In such case, it is necessary to add support for management of another application (Test) to the settings.xml file of the UIM application. The sample settingsWithTEST.xml file has been created for the purpose. The file also contains configuration for the new TEST application (along with standard supplied configuration).

To use it, replace the contents of the original settings.xml file (./webapps/UIM/WEB-INF/classes/settings.xml) - following the standard installation of ADUCID ADUCID Server Kit - with the contents of the settingsWithTEST.xml file.

After Tomcat is restarted, the UIM Role Manager can assign the new Application Manager (TEST) role. The person managing the TEST application will now have a new ‘Test Application’ tab available in the user management section. Username and role attributes can be specified on the new tab.

The list of roles is predefined in the settingsWithTEST.xml as follows:

- tomcat
- role1

7. OpenLDAP

Boot service of the OpenLDAP server is called slapd (service slapd operation).

7.1. slapd configuration files

Configuration of the LDAP server is located in the /etc/openldap directory. OpenLDAP 2.4 offers a new configuration method using LDIF in the /etc/openldap/slapd.d directory. If the directory is present, the slapd.conf file is ignored. The slapd.conf file can be converted to a slapd.d file, but not vice versa. Therefore, we recommend choosing and keeping one of the configuration methods. The configuration can be converted as follows:

```
/bin/rm -rf /etc/openldap/slapd.d
mkdir /etc/openldap/slapd.d
slaptest -f /etc/openldap/slapd.conf -F /etc/openldap/slapd.d -Q
chown -R ldap:ldap /etc/openldap/slapd.d
```


The aducid-configurator application creates an initial slapd.conf configuration file and performs a conversion to a directory. Therefore, for the administrator, the slapd.conf file is primarily of informative value, because it can be read easier than the file structure listed in the slapd.d file.

Additional information: man slapd.conf; man slapd.access; man slapd; http://www.openldap.org/doc/admin24/

7.2. Pairing attribute configuration on ADUCID Server Kit

Some applications can use “pairing” attribute for ADUCID integration. That scenario is useful for quick integration of already existing application, which we are extending of ADUCID authentication. Pairing attribute can be for example username or email. Such atribute is registered on both sides – in ADUCID and in the application too. Integrated application uses ADUCID for authentication, then it makes a request to AIM for pairing attribute. At the end, application searches its own database for user details.

In such a case it is advisable to ensure the uniqueness of the attribute in the LDAP database. This is achieved by using unique_uri directive in the configuration `/etc/openldap/slapd.conf`. For example, LDAP mail attribute uniqueness can be configured as follows:

```
loadmodule unique
overlay unique
unique_uri ldap:///?mail?sub?
```

Put these lines to the `database` section in configuration file. In ADUCID Server Kit standard installation you can simply place the configuration to end of the file. Don’t forget to convert slapf.conf file to ldif format after editing as described in the previous chapter.

7.3. LDAP client configuration file

The ldap.conf (/etc/openldap) file includes options for client programs, like ldapsearch.

Additional information: man ldap.conf

8. Apache Tomcat

Apache Tomcat listens at ports 8009, 8080, 8086. Apache Tomcat is not directly available and the communication is limited by iptables. For communication with its surroundings, the Apache HTTP Server is presented as a reverse proxy. Apache HTTP Server communicates with Apache Tomcat through the ajp protocol.

Tomcat configuration can be found in the /etc/sysconfig/tomcat6 file. Settings of the JAVA_OPTS parameters are key to system performance. The default settings in ADUCID Server Kit include the following options (wrapped for better readability).

```
JAVA_HOME=/usr/java/default
JAVA_OPTS="$JAVA_OPTS -Xmx`awk '
  /^MemTotal:/{
    m = int(($2/1024 - 128)*0.8);
    if( m > 2000 ) { m = 2000 } ;
    printf("%im\n",m);
  }' < /proc/meminfo`"
JAVA_OPTS="$JAVA_OPTS -Xms`awk '
  /^MemTotal:/{
    m = int(($2/1024 - 128)*0.5);
    if( m > 1500 ) { m = 1500 };
    printf("%im\n",m);
  }' < /proc/meminfo`"
```

The JAVA_HOME parameter specifies the java distribution used (Oracle Java).

The JAVA_OPTS parameter contains settings for JVM. By default, ADUCID Server Kit configuration includes dynamic memory size adjustment based on the total available size.

The -Xmx (maximum memory size) parameter is set as 80% of total memory less 128 MB (reserved for the system). The memory size is limited to 2 GB (32-bit limitation).

The -Xms (initial memory size) parameter is set as 50% of total memory less 128 MB (reserved for the system). The memory size is limited to 1.5 GB.

Enabled support for JMX is another difference in the configuration.

```
ADDRESS=`ip -4 addr show dev eth0 primary | awk 'BEGIN{ FS="[\t /]+"; }/^ +inet/{ print
$3; }'`
CATALINA_OPTS="-Dcom.sun.management.jmxremote=true
 -Dcom.sun.management.jmxremote.port=8086
 -Dcom.sun.management.jmxremote.ssl=false
 -Dcom.sun.management.jmxremote.authenticate=false
 -Djava.rmi.server.hostname=$ADDRESS"
```

The JMX connector allows connection via port 8086 without authentication.

In the `/etc/tomcat6/catalina.properties` configuration file, the common.loader parameter is modified to contain the path to the configuration directories. Once installation is complete, the variable setting looks like this:

```
common.loader=${catalina.base}/lib,${catalina.base}/lib/*.jar,${catalina.home}/lib,${catal
ina.home}/lib/*.jar,${catalina.base}/conf,${catalina.home}/conf
```

9. Apache HTTP Server

APACHE httpd precedes Tomcat. Its purpose is to terminate SSL connection (when using HTTPS) and filter unauthorized communication. Filtering is configured in the proxy_aducid.conf (/etc/httpd/conf.d) file.

- The proxy_aducid.conf configuration does the following:
- Hides the Apache HTTP Server version
- Prevents the use of R4 (AIM) interface by unspecified networks
- Disallows all methods other than POST for the SOAP interface
- Restricts access to URL in R3, R4, version
- Restricts access to the UIM application from unspecified networks
- Enforces the use of SSL for access to UIM
- Restricts access to AIM-proxy

10. Java

In addition to OpenJDK “system” Java, the ADUCID Server Kit also has the Oracle Java 1.6.0 installed. Oracle Java is included in the “alternatives” system. When using the “alternatives” system, this Java is set as default and Tomcat is also set to use the Oracle Java. The “alternatives” setting for Java can be displayed using the following command:

```
alternatives --display java
```

Current setting is displayed in the first few lines of the listing:

```
java - status is manual.
 link currently points to /usr/java/default/bin/java
/usr/lib/jvm/jre-1.6.0-openjdk/bin/java - priority 16000
...
...
```

Additional information: man alternatives

11. High availability

Cluster deployment provides a robust solution resistant to hardware failure. The solution does not support load distribution. Note that in a two-node cluster, the entire load must be manageable by a single cluster. Another limiting factor is that changes to LDAP database must be written in all cluster nodes.

You can create a cluster with multiple nodes. A particular solution depends on multiple factors including the load balancer used. Description of such solution is not included in this document.

The following figure shows the deployment diagram.

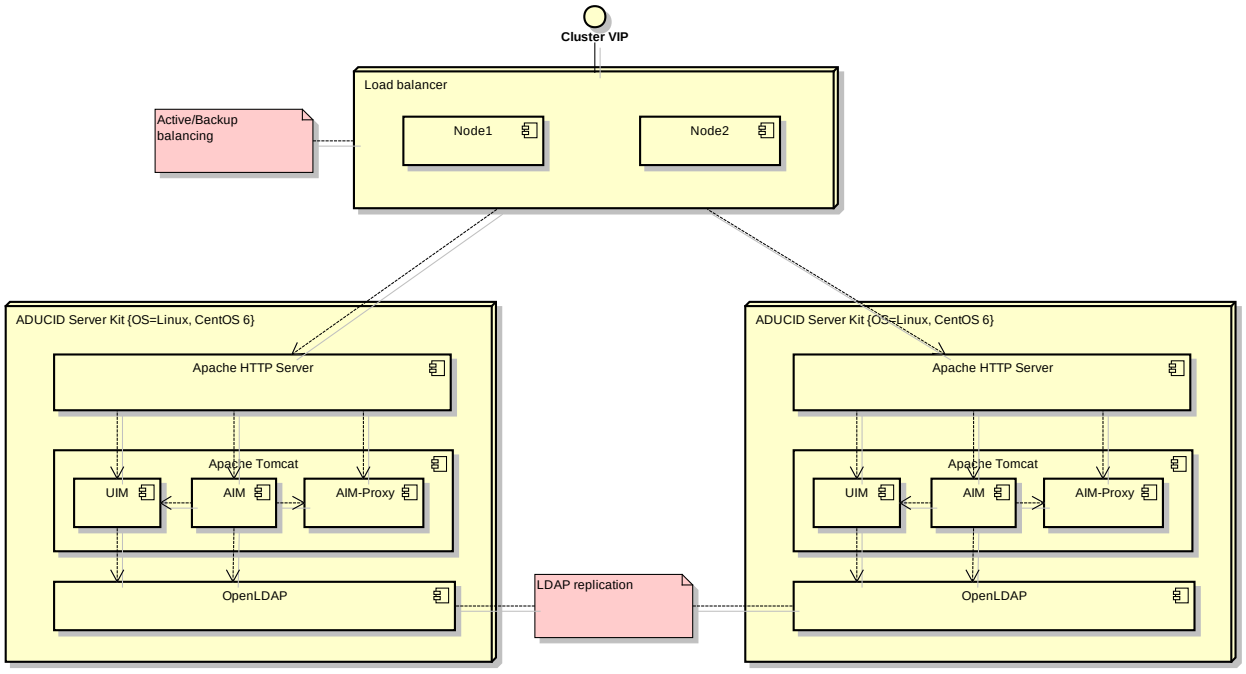

Figure 11-1 HA deployment diagram

For a two-node cluster, the load balancer must route all traffic to the functional node. It is not possible to route traffic based on client address. In ADUCID authentication, the first communication is initiated by the server part of the application (e.g. web server), and the client (e.g. browser) follows on. For the above reasons, the deployment does not rely on Tomcat servers being interconnected at individual nodes.

12. Event logs

With standard settings, the logs are stored in the /var/log directory. Individual components write into the following logs:

- AIM – /var/log/tomcat6/aim.log, /var/log/tomcat6/aim_events.log
- UIM – /var/log/tomcat6/uim.log, /var/log/tomcat6/uim_events.log
- AIM-proxy – /var/log/tomcat6/aimproxy.log, /var/log/tomcat6/aimproxy_events.log
- OpenLDAP – /var/log/messages
- Apache Tomcat – /var/log/tomcat6/catalina.out
- Apache HTTP Server – /var/log/httpd/*

Configuration of system logs is standard. Configuration of ADUCID logging components is stored in the `/usr/share/java/tomcat6/log4j.properties` file. Logging is prevented from consuming unlimited disk space - old log files are regularly deleted. By default, the ADUCID logs are set to rotate once the log size reaches 10 MB. When a log rotates, the last 10 files in the history are preserved.

If log capacity is not sufficient or you need to back up the logs, we recommend routing logging activity to a dedicated log server. You can take the following steps to configure a log server:

Define a new log appender and set proper values for SyslogHost and Facility in the /usr/share/tomcat6/lib/log4j.properties file.

```
# syslog appender
log4j.appender.SYSLOG=org.apache.log4j.net.SyslogAppender
log4j.appender.SYSLOG.SyslogHost=logserver.example.com
log4j.appender.SYSLOG.Facility=USER
log4j.appender.SYSLOG.FacilityPrinting=true
log4j.appender.SYSLOG.layout=org.apache.log4j.PatternLayout
log4j.appender.SYSLOG.layout.ConversionPattern=%d [%t] %-5p %c %x - %m%n
# syslog appender end
```

Configure the appropriate component (AIM, UIM, AIM-proxy) to use this appender with the desired logging level. Here is an example of an AIM logging component with INFO level routed to the SYSLOG appender.

```
log4j.logger.com.anect.aducid.common.logging.AIMLogger=INFO, SYSLOG
log4j.logger.com.anect.aducid.event.Log4jEventHandler=INFO, SYSLOG
```

12.1. AIM logs

The AIM component uses the following logs:

- /var/log/tomcat6/aim.log – contains support information for troubleshooting the system, level of detail of the recorded information can be changed as described in chapter 12.3.
- /var/log/tomcat6/aim_events.log – contains records of individual user events. These records can be used for an audit or as basis for accounting.

12.1.1. AIM event format

Two events are logged for each operation of the AIM application:

- Start event – the event that describes the state before performing the desired operation
- End event - the event that describes the state after performing the desired operation

Both the start and end events use the following fixed format:

```
[authId];[SPID];[current user's UDI];
[event code];[operation];[method];[session status];[authentication status];
```


The difference between the start and the end event is mainly in the event codes. Description of individual format fields:

- authId - self-explanatory (may not exist for some events, represented by the '-' character)
- SPID - service provider identifier
- Current user's UDI - apparent (may not exist for some events, represented by the '-' character)
- Operation – self-explanatory, for more information see section •. (may not exist for some events, represented by the '-' character)
- Method - related to personal objects, currently not used (represented by the '-' character)
- Session status – self-explanatory, for more information see section "6. Appendix B" of the ADUCID_Integration_manual.docx document
- Authentication status – self-explanatory, for more information see section "7. Appendix C" of the ADUCID_Integration_manual.docx document (may not exist for some events, represented by the '-' character)

1.1.1.1. List of implemented operations

Operation	Description
init	Identity initialization
reinit	Emergency repeated initialization
open	Identity use (standard authentication)
change	Identity change - manual
rechange	Identity change – if expired or allowed number of uses reached
replica	PEIG replica creation
delete	Identity deletion

12.2. UIM logs

The UIM component uses the following logs:

- /var/log/tomcat6/uim.log – contains support information for troubleshooting the system, level of detail of the recorded information can be changed as described in chapter 12.3.
- /var/log/tomcat6/uim_events.log – contains records of individual user events. These records can be used for an audit or analyze the use of this component.

In addition to the above logs, the UIM application writes the following message sequence to the standard output during start of the Tomcat application server:

```
Attempting to load ESAPI.properties via file I/O.
Attempting to load ESAPI.properties as resource file via file I/O.
Not found in 'org.owasp.esapi.resources' directory or file not readable:
/usr/share/tomcat6/ESAPI.properties
Not found in SystemResource Directory/resourceDirectory: .esapi\ESAPI.properties
Not found in 'user.home' (/usr/share/tomcat6) directory:
/usr/share/tomcat6/esapi/ESAPI.properties
Loading ESAPI.properties via file I/O failed. Exception was: java.io.FileNotFoundException
Attempting to load ESAPI.properties via the classpath.
SUCCESSFULLY LOADED ESAPI.properties via the CLASSPATH from '/ (root)' using current
thread context class loader!
Attempting to load validation.properties via file I/O.
Attempting to load validation.properties as resource file via file I/O.
Not found in 'org.owasp.esapi.resources' directory or file not readable:
/usr/share/tomcat6/validation.properties
Not found in SystemResource Directory/resourceDirectory: .esapi\validation.properties
Not found in 'user.home' (/usr/share/tomcat6) directory:
/usr/share/tomcat6/esapi/validation.properties
Loading validation.properties via file I/O failed.
Attempting to load validation.properties via the classpath.
```



```
SUCCESSFULLY LOADED validation.properties via the CLASSPATH from '/ (root)' using current
thread context class loader!
```

The message sequence is an integral part of the UIM application start.

12.2.1. UIM event format

Two events are logged for each operation of the UIM application:

- Start event – the event that describes the moment/state before performing the desired operation
- End event - the event that describes the moment/state after performing the desired operation

The start event has the following fixed format:

```
[ID and component version];[category];[event code];[current user’s UDI];[authId];-;-
;[event message body];
```

The end event has the following fixed format:

```
[ID and component version];[category];[event code];[current user’s UDI];[authId];[event
outcome, SUCCESS or FAILURE];[reason for failure (only in the event of FAILURE)];[event
message body];
```

There is a clear difference between the two events - fields that provide information on the operation’s outcome are not defined for the start event. Description of individual format fields:

- ID and component version – self-explanatory
- Category - event category; the following categories are defined:
  - Security (SEC)
  - Application (APP)
  - System (SYS)
  - Monitoring (MON)
- Current user’s UDI - self-explanatory (may not exist for some events, non-existence represented by the ‘-‘ character)
- authId - self-explanatory (may not exist for some events, non-existence represented by the ‘-‘ character)
- Event outcome – “SUCCESS” value in case of success, “FAILURE” value in case of failure; the event outcome is written only to the end event
- Reason for failure – the reason why the operation failed; written only if the outcome of the operation has the “FAILURE” value; can be any text
- Event message body – any text ideally presenting data that are part of the operation at runtime; if it is necessary to preserve additional structure (i.e. free text does not suffice), a predefined structure is inserted into this field

12.3. Setting the logging level for AIM and UIM

It is possible to change UIM and AIM log level during runtime. There is aducid-loglevel command for this purpose – see chapter 14.1

12.4. AIM-Proxy logs

The AIM-Proxy component uses the following logs:

- /var/log/tomcat6/aimproxy.log – does not contain any information; the file is ready for future use.
- /var/log/tomcat6/aimproxy_events.log – does not contain any information; the file is ready for future use.

12.5. Informative messages at application shutdown

When shutting down the AIM, AIM-Proxy or UIM application, the following messages may show in the standard output (standard output is either a console or Apache Tomcat log files with `catalina.*` prefix):

```
SEVERE: A web application created a ThreadLocal with key of type [the rest is not
substantial]
```

Despite the logging level (SEVERE logging level indicates a major error), these are unimportant messages that inform the developer about potential risks when working with objects in memory. These messages do not affect the functionality of applications. Tens of these messages may appear.

12.6. OpenLDAP

OpenLDAP does not log any events. Enabling logging is not recommended, as it could affect system performance.

12.7. Apache Tomcat

The Apache Tomcat application uses the following main logs:

catalina.[date].log - record of the application server's standard output

localhost.[date].log - recorded information related to default host (localhost)

12.8. Apache HTTP

Standard configuration is applied following installation of the package. It defines the following logs:

- access_log - http access log
- error_log - log of errors that occurred while handling http requests
- ssl_access_log - https access log
- ssl_error_log - log of errors that occurred while handling https requests
- ssl_request_log - this log is similar to the ssl_access_log; contains additional information on SSL (encoding used, etc.)

13. Attribute sets

ADUCID allows you to create attribute sets for individual applications that are consequently available to these applications when working with users' personal data. Attributes define the name of the attribute and its mapping to an attribute within LDAP. They also define whether the attribute in the given set is for reading or writing.

A new attribute set can only be defined by modifying the LDAP database. Recommended steps: prepare an ldif file and add it to the database.

By default, the attribute sets are located in ou=rights,ou=config. A separate container is created for each attribute set. With regards to ADUCID, the organization of objects in LDAP is not important. It is recommended to respect the organization though as it will enable you to review objects easily.

The following example shows the ldif file for an attribute set named "MY" that contains one attribute named "TEST" (this attribute will be stored in LDAP under the "pager" attribute). When using the "MY" attribute set, the "TEST" attribute will be read-only. The purpose of permissions in attribute sets is to protect them from unintentional error, rather than from an attack. There is no check to verify what set the application uses.

Attribute sets allow applications to share attributes in LDAP under various names ("color" and "colour" attributes may lead to the same attribute in LDAP). They may also serve to separate attributes where one

attribute name is mapped to different LDAP attributes in LDAP. Two applications may then use an attribute with the same name while having a separate value for each of the applications.

First, create a file with a basic container for “MY” set. Parts that can be changed according to your preference are highlighted in red.

```
dn: aducidApplication=MY,ou=rights,ou=config,dc=example,dc=com
objectClass: aducidApplicationContainer
aducidApplication: MY
```

Then define individual attributes of the set and permissions for working with this set.

```
dn: aducidAlias=TEST,aducidApplication=MY,ou=rights,ou=config,dc=example,dc=com
aducidAlias: TEST
objectClass: aducidApplicationMap
aducidApplication: MY
aducidRightWrite: FALSE
aducidRightRead: TRUE
aducidAttribute: pager
```

All objects can be placed in a single file separated by an empty line.

By default, the aducidUser class derived from inetOrgPerson is used in LDAP. This determines what LDAP attributes can be used.

The entire file in our example may look like this:

```
dn: aducidApplication=MY,ou=rights,ou=config,dc=example,dc=com
objectClass: aducidApplicationContainer
aducidApplication: MY
structuralObjectClass: aducidApplicationContainer

dn: aducidAlias=TEST,aducidApplication=MY,ou=rights,ou=config,dc=example,dc=com
aducidAlias: TEST
objectClass: aducidApplicationMap
aducidApplication: MY
aducidRightWrite: FALSE
aducidRightRead: TRUE
aducidAttribute: pager
structuralObjectClass: aducidApplicationMap
```

You can add the file to LDAP using the following command line commands:

```
service slapd stop
slapadd <MY.ldif
chown ldap:ldap /var/lib/ldap/*
service slapd start
service tomcat6 restart
```

Tip: The slapadd expects unix-type end-of-line characters.

14. Command line tools

14.1. aducid-loglevel

This command changes the logging level of AIM, UIM and AIMProxy at runtime. You can enter the command without parameters to display current settings.


```
[root@orangebox ~]# aducid-loglevel
Log level for AIMProxyLogger is INFO
Log level for UIMLogger is INFO
Log level for AIMLogger is INFO
```

Use the `-h` or `--help` parameter to display a simple user guide. Use the `--set` parameter to set the logging level for components.

```
 [root@orangebox ~]# aducid-loglevel --set debug --component aim
Log level for AIMLogger is now DEBUG
```

Available components:

- aim
- uim
- aimproxy
- all (set for all components)

Available logging level values:

- debug
- info
- warn
- error
- fatal
- all
- off

14.2. aducid-user

This command prints information about user from LDAP. The tool prints all object connected with user – user itself, proxy object and objects of identities. It prints simple help using the `-h` or `--help` switch.

```
[root@shaim ~]# aducid-user -h
usage: aducid-user [options]
  -a|--attr attribute        find user according some ldap attribute
  -v|--value attributeValue  value of searched attribute [*].
                             for example aducid-user -a cn -v "Joe*"
  -r|--ref refferal          attribute used for pairing user information
                             [aducidGUID].
  -u|--udi userDatabaseIndex find user with given userDatabaseIndex
```

It is possible to find user according userDatabaseIndex (`--udi`) or according user's atrribute (`--attr`). It is necessary to know exact value in case of lookup with udi. It is possible to use * character in case of attribute search. Behavior of wild character depends also on LDAP server settings. This tool only creates appropriate LDAP search filter.

```
aducid-user --attr sn --val "J*"
```

This example searches for all users with surname beginning with "J". LDAP filter is (sn=J*). Output can look like this:

```
-------------------- user data --------------------
dn: aducidGUID=420973a7-e4ea-4e55-a52a-dc4cbe075eca,ou=people,dc=aducid,dc=eu
aducidGUID: 420973a7-e4ea-4e55-a52a-dc4cbe075eca
objectClass: aducidUser
mail: email@aducid.eu
sn: Joe
cn: Joe
```



```
-------------------- user data --------------------
                          |
                          |
------------------ proxy object -------------------
dn: aducidUserProxyId=c0af1da3-171f-4ae4-b3da-d5a0d40a4a81,ou=proxies,dc=aducid,dc=eu
objectClass: aducidUserProxy
aducidUserProxyId: c0af1da3-171f-4ae4-b3da-d5a0d40a4a81
aducidAppReferral: aducidGUID#420973a7-e4ea-4e55-a52a-dc4cbe075eca
aducidAuthorized: FALSE
aducidUdi: 611877989
aducidFormId: 72c5586a-d3b9-4796-a64d-c1e93a4a81d6
aducidDateLinked: 1360842014
aducidRole: UIM_U
------------------ proxy object -------------------
|
|
|  ----------------- identity ---------------------
+--| dn: aducidUid=R8rUi0J2OCM\3D,ou=identities,dc=aducid,dc=eu
   | objectClass: aducidIdentity
   | aducidSpid: aim-spid
   | aducidUidSP: i0J2OCM=
   | aducidUid: R8rUi0J2OCM=
   | aducidUdi: 611877989
   | aducidSerializedData:  - not printable -
   | aducidIdentityActive: TRUE
   ----------------- identity ---------------------
```

Command requires root access (for reading ADUCID.properties file).

14.3. aducid-role

This command is used to retrieve or set user roles in UIM. It prints simple Help, by using the `--help` switch.

```
[root@orangebox ~]$ aducid-role --help
usage: aducid-role [options]
  -u|--udi userDatabaseIndex    find user with given userDatabaseIndex
  -a|--add ROLE                 add role ROLE to user with given
                                userDatabaseIndex
  -d|--del|--delete ROLE        delete role ROLE from user with given
                                userDatabaseIndex
  -A|--attr|--attribute ATTR    find user by attribute
  -v|--value VAL                with value
  -r|--ref referral             referral attribute [aducidGUID]
```

It is possible to find a user by using **userDatabaseIndex** (`--udi`) or according to a user's attribute (`--attr`). It is necessary to know the exact value in case of lookup with udi. It is possible to use the * wildcard, in the case of an attribute search. For example:

```
aducid-role --attr sn --val "J*"
```

The previous example will print roles of all users with surname beginning with "J".

The parameters`--add` and `--del` can be used multiple times. This makes it possible add or remove multiple roles at once. For example:

```
aducid-role --attr sn --val "Smith" --add UIM_SM --add UIM_RM
```


The previous example sets the roles UIM_SM and UIM_RM to the user with the surname “Smith”. The roles defined in UIM are:

- UIM_SM—Security manager
- UIM_RM—Role manager
- UIM_UM—User manager
- UIM_U—User

15. System monitoring

You can monitor the system using SNMP. SNMP is disabled by default. After enabling SNMP, follow these mandatory steps:

- Install the aducid-snmp package (included on the supplied DVD)
- Allow SNMP communication in iptables (udp/161)
- Configure net-snmpd
- Ensure net-snmpd is set to start automatically (`chkconfig snmpd on; service snmpd start`)
- Configure your monitoring system

Monitoring of the following metrics is recommended:

- CPU load
- Number of connections
- Disk space
- Test aducidwebservices
- Test aducidldap

The snmpd.conf configuration file may look like this:

```
com2sec notConfigUser  default       public
group   notConfigGroup v1           notConfigUser
group   notConfigGroup v2c           notConfigUser
view    systemview    included   .1.3.6.1.2.1.1
view    systemview    included   .1.3.6.1.2.1.25.1.1
access  notConfigGroup "" any noauth exact systemview none none
syslocation Server Room 42
syscontact Admin <Admin@example.com>
pass .1.3.6.1.4.1.4413.4.1 /usr/bin/ucd5820stat
view    allview       included   .1
com2sec aducidUser  localhost  aducid
com2sec aducidUser  10.10.10.10   aducid
com2sec aducidUser  10.10.10.11   aducid
group   aducidGroup    v1            aducidUser
group   aducidGroup    v2c           aducidUser
access aducidGroup "" any noauth exact allview none none
extend aducidwebservices /usr/sbin/aducid-webapps-test
extend aducidldap        /usr/sbin/aducid-ldap-test
load 12 30 50
disk / 500000
```

In this example, the 10.10.10.10 and 10.10.10.11 servers with “aducid” community name are allowed to perform monitoring. The net-snmpd property is used in addition to standard MIB. It allows the use of scripts that enable MIB to include additional information. The MIB begings at OID 1.3.6.1.4.1.8072.1.3.2.3.1

You can read a numeric value of the test result:

- 0 - OK
- 1 - warning

- 2 - error
- 4 - unknown status

You can also get a text representation of the test result.

Test aducidwebservices

```
Numeric result
1.3.6.1.4.1.8072.1.3.2.3.1.4.17.97.108.117.99.105.100.119.101.98.115.101.114.118.105.99.10
1.115
Text result
1.3.6.1.4.1.8072.1.3.2.3.1.1.17.97.108.117.99.105.100.119.101.98.115.101.114.118.105.99.10
1.115
```

Test aducidldap

```
Numeric result
1.3.6.1.4.1.8072.1.3.2.3.1.4.10.97.108.117.99.105.100.108.100.97.112
Text result
1.3.6.1.4.1.8072.1.3.2.3.1.1.10.97.108.117.99.105.100.108.100.97.112
```

Additional information: man snmpd.conf

16. Backup and restore

The best practice is to back up the entire virtual machine using a virtual infrastructure, like VMware Consolidated Backup. A machine backed up this way can be easily restored, thus minimizing downtime.

17. Appendices

This section contains examples of configuration files.

17.1. ADUCID.properties

```
#AIM
AIM_host=http://aim.example.com:80
SPID=aim.example.com-10.11.12.13
ICON=/usr/share/pixmaps/aducid.png
IMAGE_DIR=/usr/share/pixmaps/aim
DISPLAY_NAME=ADUCID AIM Server
AUTH_LIFE_TIME=20
AUTHID_LENGTH=8
IMPLICIT_ACTIVITY=open
START_TIMEOUT=30
PROCESS_TIMEOUT=30
BINDING_TIMEOUT=30
BINDING_CONTROL_MODE=0
HOME_URL=http://www.aducid.eu/UIM
# MEETING_ROOM_ENTER_TIME_LOCK=150
# MEETING_ROOM_CONFIRM_TIME_LOCK=150
# MEETING_ROOM_MAX_REPLICA_COUNT=10
# MEETING_ROOM_MIN_PARALLEL_LF_SUCCESS=1
# MEETING_ROOM_MAX_PARALLEL_LF_FAILURE=3
# MEETING_ROOM_MAX_FAILURE_COUNT=12
LF_LOCK_DELAY=30
LF_LOCK_BUNDLE_COUNT=3
LF_BUFFER_TIME=30
LF_BUFFER_REFRESH_ENABLED=true
# PAYMENT_ASYMETRIC_KEY_ALGORITHM=RSA
```

```
# PAYMENT_ASYMETRIC_KEY_SIZE=1024
# PAYMENT_SIGNATURE_ALGORITHM=SHA1withRSA
# PAYMENT_HASH_ALGORITHM=SHA-1

#ldap
ldap.url=ldap://localhost:389
ldap.username=cn=uimmanager,ou=config,dc=example,dc=com
ldap.password=SFseZdsvEW343422ed
ldap.baseDN=dc=example,dc=com
ldap.rightConfigDN=ou=rights,ou=config
ldap.securityProfileDN=ou=profiles,ou=config
ldap.identityDN=ou=identities
ldap.userDN=ou=people
ldap.userProxyDN=ou=proxies
ldap.ilidDN=ou=ilids
ldap.roleDN=ou=roles,ou=config
ldap.applicationDN=ou=applications,ou=config
ldap.permissionDN=ou=permissions,ou=config
ldap.ILSecurityProfileDN=ou=ilprofiles,ou=config
ldap.legacyProfileDN=ou=legacyprofiles,ou=config
ldap.userAttrSpecDN=ou=applications,ou=config

#database
db.path=file:/opt/aducid/db
db.name=aducid
db.username=SA
#db.password=ldap.password
```

17.2. UIM.properties

```
referral=aducidGUID
defaultKeyValidityHours=48
AIM_proxy_url=https://aim.example.com:8443/AIM-proxy/startAuth

activationKeyLength=16

enableLBSecurityProfiles=false
```

17.3. slapd.conf

```
########################################################################
# schema definition
########################################################################

include         /etc/openldap/schema/core.schema
include         /etc/openldap/schema/cosine.schema
include         /etc/openldap/schema/inetorgperson.schema
include         /etc/openldap/schema/nis.schema
include         /etc/openldap/schema/aducid.schema
include         /etc/openldap/schema/aducid-user.schema

########################################################################
# running params
########################################################################

allow bind_v2
pidfile         /var/run/openldap/slapd.pid
argsfile        /var/run/openldap/slapd.args

########################################################################
# Rights definition
########################################################################
```

```
access to attrs=userPassword
 by self =rwx stop
 by anonymous =x stop
 by * =0 stop

access to attrs=aducidSerializedData
 by dn.exact="cn=uimmanager,ou=config,dc=example,dc=com" =rscxw stop
 by * =0 stop

access to dn.subtree="ou=people,dc=example,dc=com"
 by dn.exact="cn=uimmanager,ou=config,dc=example,dc=com" =rscxw stop
 by * =0 break

access to dn.subtree="ou=config,dc=example,dc=com"
 by dn.exact="cn=uimmanager,ou=config,dc=example,dc=com" =rscxw stop
 by * =0 stop

access to dn.subtree="ou=identities,dc=example,dc=com"
 by dn.exact="cn=uimmanager,ou=config,dc=example,dc=com" =rscxw stop
 by * =0 stop

access to dn.subtree="ou=proxies,dc=example,dc=com"
 by dn.exact="cn=uimmanager,ou=config,dc=example,dc=com" =rscxw stop
 by * =0 stop

access to dn.subtree="ou=ilids,dc=example,dc=com"
 by dn.exact="cn=uimmanager,ou=config,dc=example,dc=com" =rscxw stop
 by * =0 stop

access to dn.subtree="dc=example,dc=com"
 by * =rsc

#######################################################################
# ldbm and/or bdb database definitions
#######################################################################

database        bdb
suffix          "dc=example,dc=com"
rootdn          "cn=manager,dc=example,dc=com"
#
# see slappasswd for rootpw
# rootpw                {crypt}ijFYNcSNctBYg

directory       /var/lib/ldap

index objectClass                       eq,pres
index ou,cn,mail,surname,givenname      eq,pres,sub
index uidNumber,gidNumber,loginShell    eq,pres
index uid,memberUid                     eq,pres,sub
index nisMapName,nisMapEntry            eq,pres,sub
index aducidUid,aducidSpid,aducidUdi    eq,pres
index aducidProfileActive               eq,pres
index aducidProfileName                 eq,pres,sub
index aducidGUID                        eq,pres
index aducidUserProxyId                 eq,pres
index aducidApplication,aducidAttribute eq,pres
index aducidAppReferral                 eq,pres
index aducidUidSP                       eq,pres
index aducidActivationKey               eq,pres
```

17.4. ldap.conf

```
BASE dc=example,dc=com
```

17.5. proxy_aducid.conf

```
#
# anonymize
#
ServerTokens ProductOnly
KeepAlive On
MaxClients 100

#
# ---------------------------------------- AIM R3
# R3 is secure interface, secured "by design"
# it doesn't need protection
#
<Location ~ "^/AIM/services/R3$">
    <Limit POST>
        Order Deny,Allow
        Allow from all
    </Limit>
    <LimitExcept POST>
        Order Deny,Allow
        Deny from all
    </LimitExcept>
</Location>
<Location ~ "^/AIM/services/R3.+">
    Order Deny,Allow
    Deny from all
</Location>
<Location ~ "^/AIM/services/R3/ping$">
    <Limit GET>
        Order Deny,Allow
        Allow from all
    </Limit>
    <LimitExcept GET>
        Order Deny,Allow
        Deny from all
    </LimitExcept>
</Location>

ProxyPass /AIM/services/R3 ajp://localhost:8009/AIM/services/R3 retry=1 acquire=10000
timeout=15 ttl=5

#
# ---------------------------------------- AIM R4
#
<Location ~ "^/AIM/services/R4$">
    <Limit POST>
        Order Deny,Allow
        Deny from all
        Allow from X.X.X.X/Y
        Allow from 127.0.0.1
    </Limit>
    <LimitExcept POST>
        Order Deny,Allow
        Deny from all
    </LimitExcept>
</Location>
<Location ~ "^/AIM/services/R4.+">
    Order Deny,Allow
```

```
    Deny from all
</Location>

ProxyPass /AIM/services/R4 ajp://localhost:8009/AIM/services/R4 retry=1 acquire=10000
timeout=15 ttl=5
#
# ---------------------------------------- AIM R7
#
<Location ~ "^/AIM/services/R7$">
    <Limit POST>
        Order Deny,Allow
        Allow from all
    </Limit>
    <LimitExcept POST>
        Order Deny,Allow
        Deny from all
    </LimitExcept>
</Location>
<Location ~ "^/AIM/services/R7.+">
    Order Deny,Allow
    Deny from all
</Location>

ProxyPass /AIM/services/R7 ajp://localhost:8009/AIM/services/R7 retry=1 acquire=10000
timeout=15 ttl=5

#
# ---------------------------------------- AIM global
#
ProxyPass /AIM/version ajp://localhost:8009/AIM/version retry=1 acquire=10000 timeout=15
ttl=5

#
# ---------------------------------------- UIM
#
<Location /UIM>
    Order Deny,Allow
    Deny from all
    Allow from X.X.X.X/Y
    Allow from 127.0.0.1

    RewriteEngine On
    RewriteCond %{HTTPS} off
    RewriteRule .* https://%{HTTP_HOST}%{REQUEST_URI}
</Location>

ProxyPass /UIM ajp://localhost:8009/UIM retry=1 acquire=10000 timeout=150 ttl=5

#
# ---------------------------------------- AIM proxy
#
<Location ~ "^/AIM-proxy/(version|open)">
    <LimitExcept GET>
        Order Deny,Allow
        Deny from all
    </LimitExcept>
    <Limit GET>
        Order Deny,Allow
        Deny from all
        Allow from X.X.X.X/Y
        Allow from 127.0.0.1
    </Limit>
</Location>

<Location ~ "^/AIM-proxy/(checkStatus|process)">
```

```
    <LimitExcept GET>
        Order Deny,Allow
        Deny from all
    </LimitExcept>
    <Limit GET>
        Order Deny,Allow
        Allow from all
    </Limit>
</Location>

ProxyPass /AIM-proxy ajp://localhost:8009/AIM-proxy retry=1 acquire=10000 timeout=15 ttl=5

#
# redirect "/" request to uim
#
RedirectMatch ^/$ UIM/
```

17.6. snmpd.conf

```
com2sec notConfigUser  default       public
group   notConfigGroup v1           notConfigUser
group   notConfigGroup v2c           notConfigUser
view    systemview    included   .1.3.6.1.2.1.1
view    systemview    included   .1.3.6.1.2.1.25.1.1
access  notConfigGroup "" any noauth exact systemview none none
syslocation Server Room 42
syscontact Admin <Admin@example.com>
pass .1.3.6.1.4.1.4413.4.1 /usr/bin/ucd5820stat
view    allview       included   .1
com2sec aducidUser  localhost  aducid
com2sec aducidUser  10.10.10.10   aducid
com2sec aducidUser  10.10.10.11   aducid
group   aducidGroup    v1            aducidUser
group   aducidGroup    v2c           aducidUser
access aducidGroup "" any noauth exact allview none none
extend aducidwebservices /usr/sbin/aducid-webapps-test
extend aducidldap        /usr/sbin/aducid-ldap-test
load 12 30 50
disk / 500000
```

18. Abbreviations

Below is a summary of used abbreviations and their meaning:

AIM **ADUCID® Identity Machine**

OS Operating system

NTP **Network Time Protocol**

DNS Domain Name System

VM **Virtual Machine**

LDAP Lightweight Directory Access Protocol

DN **Distinguished Name**

CA Certification Authority

ADUCID®

ADUCID® is a new authentication system providing identity services and infrastructures. Based on new ideas, rules, procedures and implementations, ADUCID® establishes an identification and authentication framework, within which a unified authentication method can function and be supported.

The main purpose of ADUCID® is to provide identification and authentication services in the cybernetic world of ICT systems using an ADUCID® secure authentication layer.

ADUCID® provides:

- Identity services
- Secure authentication services
- Essential infrastructure for these services

PEIG®

PEIG® (Personal Electronic Identity Guardian) is a device that can fully manage identities of its user. Under the user identity it also provides automatic authentication between the client application (used by the user) and the server part of the target application (that the user is accessing).

AIM

ADUCID® Identity Machine - implements ADUCID® server functionality. It performs all ADUCID® operations and provides access to user data stored along with identities in the LDAP database.

Using standard network interface (web services), it provides target applications with services related to identity administration. AIM contains administrator and user graphic interface (called UIM). AIM can also provide authorization services (including administration of authorization attributes of the relevant identities).

AIM-proxy

Specialized module for web applications used to communicate with the client web browser upon authentication of HTML applications. This component enables ADUCID® to login to UIM without modifying the browser (redirect login).

PEIG-proxy

Specialized software communication module that connects PEIG® Core to a client target application and AIM. It also functions as an application firewall to protect PEIG® Core. It must be run on the same computer as the client part of the target application.

19. Literature

[1] *ADUCID Architecture*

[2] *ADUCID Integration Manual*

[3] *ADUCID Server Kit - Installation Guide*

[4] *UIM - Administration Guide*

[5] *ADUCID Binding*

