



Installazione Shibboleth IDP 2.4.0 per Debian Linux

28 Gennaio 2015

Autore: Marco Malavolti

Credits: Switch AAI, Shibboleth

Indice generale

1) Introduzione.....	3
2) Software da installare.....	3
3) Richiedere il certificato per l'IDP.....	4
4) Installare Tomcat 7, openjdk-7-jdk, openssl, Apache 2, unzip e ntp.....	5
5) Installare Shibboleth Identity Provider 2.4.0.....	7
6) Configurare Tomcat 7.....	9
7) Configurare Shibboleth Identity Provider 2.4.0.....	13

1 Introduzione

Questo documento ha lo scopo di guidare l'utente nell'installazione di un IdP Shibboleth 2.4.0 con Tomcat 7 e Apache 2 come front-end.

2 Software da installare

- openssl
- ca-certificates
- ntp
- openjdk-7-jdk
- tomcat7
- apache2
- unzip
- expat (per il parsing di xml)
- idp (<http://www.shibboleth.net/downloads/identity-provider/latest/>)

3 Richiedere il certificato per l'IDP

- 1) In linea con le **specifiche tecniche** della Federazione IDEM è necessario installare sulla porta 443 un certificato rilasciato da una CA riconosciuta. All'interno della comunità GARR è attivo il servizio di rilascio certificati server denominato **TCS** (TERENA Certificate Service). La caratteristica dei certificati TCS è quella di essere emessi da una CA commerciale che nello specifico consiste in **COMODO CA**.
 - L'elenco delle organizzazioni presso le quali il servizio TCS è già attivo è disponibile in <https://ca.garr.it/TCS/tab.php>
 - Se il servizio non fosse ancora attivo presso la vostra organizzazione è possibile contattare **GARR Certification Service** per avviare il procedimento di attivazione (e-mail a garr-ca@garr.it)
- 2) Per generare una richiesta di certificato seguire le istruzioni suggerite nelle pagine di documentazione TCS (https://ca.garr.it/TCS/doc_server.php)
- 3) Le richieste di certificato devono essere inviate ai **referenti TCS** presenti nella vostra organizzazione (denominati Contatti Amministrativi TCS). Per conoscere i nomi dei Contatti Amministrativi nominati all'interno del vostro Ente inviare una mail di richiesta a garr-ca@garr.it

4 Installare Tomcat 7, openjdk-7-jdk, openssl, Apache 2, unzip e ntp¹

1) Installare i seguenti pacchetti Debian:

- `sudo apt-get install openjdk-7-jdk ca-certificates openssl ntp tomcat7 apache2 unzip`

2) Aggiungere le seguenti variabili all'environment

- `sudo vim /etc/environment`

```
CATALINA_HOME=/usr/share/tomcat7
CATALINA_OUT=/var/log/tomcat7/catalina.out
TOMCAT_HOME=/var/lib/tomcat7
TOMCAT_LOG_DIR=/var/log/tomcat7
JAVA_ENDORSED_DIRS=/usr/share/tomcat7/endorsed
IDP_HOME=/opt/shibboleth-idp
IDP_SRC=/usr/local/src/shibboleth-identityprovider-2.4.0
IDP_LOG=/opt/shibboleth-idp/logs/idp-process.log
```

3) Fare Logout e Login per attuare i cambiamenti all'environment della macchina

4) Avviare Tomcat:

- `service tomcat7 start`

¹ per ubuntu 10.04 e superiori

Passo Facoltativo) Amministrare Tomcat da <http://localhost:8080/manager/html>:

1. Acquisire i permessi di root
 - `sudo su -`
2. Installare il pacchetto “tomcat7-admin”
 - `apt-get install tomcat7-admin`
3. Modificare “`$CATALINA_BASE/conf/tomcat-users.xml`” per aggiungere gli utenti:

```
<tomcat-users>
  ...
  <role rolename="manager"/>
  <role rolename="administrator"/>

  <user username="Admin" password="password_Ammministratore"
        roles="admin,manager"/>

  <user username="Manager" password="password_Manager"
        roles="manager"/>
</tomcat-users>
```

5 Installare Shibboleth Identity Provider 2.4.0

1) Acquisire i permessi di ROOT:

- `sudo su -`

2) Spostarsi nella cartella `/usr/local/src`:

- `cd /usr/local/src`

3) Scaricare lo Shibboleth IdP 2.4.0:

- `wget http://shibboleth.net/downloads/identity-provider/latest/shibboleth-identityprovider-2.4.0-bin.zip`
- `unzip shibboleth-identityprovider-2.4.0-bin.zip`
- `cd shibboleth-identityprovider-2.4.0`

4) Modificare il file `$IDP_SRC/src/main/webapp/WEB-INF/web.xml` sostituendo **#your IP range#** con il proprio [CIDR](#):

```
<!-- Servlet for displaying IdP status. -->
<servlet>
  <servlet-name>Status</servlet-name>
  <servlet-class>
    edu.internet2.middleware.shibboleth.idp.StatusServlet
  </servlet-class>

  <!-- Space separated list of CIDR blocks allowed to access the status page -->
  <init-param>
    <param-name>AllowedIPs</param-name>
    <param-value>
      127.0.0.1/32 ::1/128 #your IP range#
    </param-value>
  </init-param>

  <load-on-startup>2</load-on-startup>
</servlet>
```

Questo attiverà la pagina `https://idp.example.org/idp/status` con cui visualizzare informazioni aggiuntive sull'IdP.

5) Installare l'IdP:

```
sh install.sh
```

[scegliere il FQDN per l'Identity Provider (Predefinito: "idp.example.org") e annotarsi per precauzione la password del suo keystore (Es. 123456) che **NON** servirà con questa modalità di installazione dello Shibboleth IdP]

(Lasciare come percorso di installazione quello predefinito `"/opt/shibboleth-idp"`)

- 6) Copiare le librerie di Xerces (Java parser for XML) e di Xalan (Xalan è un XSLT processor per trasformare documenti XML in documenti HTML, testo, o altri documenti XML) in \$TOMCAT_HOME:
 - `cp -r $IDP_SRC/endorsed/ $CATALINA_HOME`
- 7) Modificare i permessi per abilitare l'utente tomcat7 ad accedere alle directory dell'IdP:
 - `chown tomcat7 $IDP_HOME/logs/`
 - `chown tomcat7 $IDP_HOME/metadata/`
 - `chown tomcat7 $IDP_HOME/credentials/`
- 8) E i permessi sul certificato e la chiave creati dall' `install.sh`:
 - `chmod 400 $IDP_HOME/credentials/idp.key`
 - `chmod 644 $IDP_HOME/credentials/idp.crt`
 - `chown tomcat7 $IDP_HOME/credentials/idp.key`
 - `chown tomcat7 $IDP_HOME/credentials/idp.crt`

6 Configurare Tomcat 7

- 1) Acquisire i permessi di ROOT:
 - `sudo su -`
- 2) Modificare "server.xml":
 - `vim $TOMCAT_HOME/conf/server.xml`aggiungendo il seguente connettore:

```
<Connector port="8009"
  protocol="AJP/1.3"
  redirectPort="443"
  address="127.0.0.1"
  enableLookups="false"
  tomcatAuthentication="false" />
```

E commentando il `<Connector port=8080 ...`

- 3) Modificare `/etc/default/tomcat7`:
 - `vim /etc/default/tomcat7`
 - a) Decomentare e modificare la `JAVA_HOME` come segue:
 - `JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64`
 - b) Decomentare e modificare le `JAVA_OPTS` come segue:
 - `JAVA_OPTS="-Djava.awt.headless=true -XX:+DisableExplicitGC -XX:+UseParallelOldGC -Xms256m -Xmx2g -XX:MaxPermSize=512m"`

(In questo modo si configura la memoria della JVM per esaudire l'IdP Web Application. Il valore per la memoria usata dipende dalla memoria fisica del server. Impostare `Xmx` (massimo heap space a disposizione della JVM) ad almeno 2GB e `XX:MaxPermSize` a 512 MB.)
 - c) Decomentare la riga `#AUTHBIND=no` e modificarla in `AUTHBIND=yes` in modo da permettere a TOMCAT di poter usare le porte **inferiori** a 1024. (necessario per usare la 443)
- 4) Posizionare la chiave privata utilizzata per la creazione del Certificato del Server (**key-server.pem**) e il certificato del server che è stato rilasciato (**cert-server.pem** - es.: `cert-9999-prova.lab.test.it.pem`) nella cartella, da voi creata precedentemente, `“$IDP_HOME/credentials”`.
- 5) Depositare l' IdP WAR file, localizzato in `$IDP_HOME/war/` usando un context deployment fragment:

La normale procedura per il deploying delle Web Application in Tomcat è attuata mediante la copia del file WAR nella cartella `webapps/` di Tomcat.

Tuttavia, quando questa procedura viene eseguita, Tomcat espande il WAR file (ottenendo così il file `idp/` nella cartella `webapps/` ma senza cancellare il file WAR) e carica la nuova

versione dell'applicazione in "work/Catalina/localhost/". Questo può causare l'utilizzo di una precedente versione del WAR anche se viene copiata una versione nuova nella giusta posizione (webapps/).

Per ovviare a questo inconveniente, viene raccomandato di usare un context deployment fragment. Questo significa che si userà un piccolo pezzo di XML per dire a Tomcat dove andare a prendere il WAR e fornire qualche proprietà da usare quando Tomcat caricherà l'applicazione.

- 6) Creare e Modificare il file "idp.xml":

```
sudo vim /etc/tomcat7/Catalina/localhost/idp.xml
```

e copiarvi dentro il seguente pezzo di codice:

```
<Context docBase="/opt/shibboleth-idp/war/idp.war"
  privileged="true"
  antiResourceLocking="false"
  antiJARLocking="false"
  unpackWAR="false"
  swallowOutput="true" />
```

- 7) Abilitare il modulo "SSL" e il modulo "proxy_ajp" di Apache2:

- sudo a2enmod ssl proxy_ajp ; service apache2 restart

- 8) Creare un copia del file "default-ssl" in "/etc/apache2/sites-available" nominata "idp-ssl" con le seguenti modifiche:

```
<VirtualHost _default_:443>
  ServerName idp.example.org:443
  ServerAdmin admin@example.org

  DocumentRoot /var/www

  <Proxy ajp://localhost:8009>
    Allow from all
  </Proxy>

  ProxyPass /idp ajp://localhost:8009/idp retry=5
  ProxyPassReverse /idp ajp://localhost:8009/idp retry=5

  SSLEngine On
  SSLCipherSuite HIGH:MEDIUM:!aNULL:!MD5
  SSLProtocol all -SSLv2 -SSLv3

  SSLCertificateFile /opt/shibboleth-idp/certs/cert-server.pem
  SSLCertificateKeyFile /opt/shibboleth-idp/certs/key-server.pem
  SSLCertificateChainFile /opt/shibboleth-idp/certs/Terena-Chain.pem

  BrowserMatch "MSIE [2-6]" \
    nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0
  # MSIE 7 and newer should be able to use keepalive
  BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown

</VirtualHost>
```

```
<VirtualHost _default_:8443>
  ServerName idp.example.org:8443
  ServerAdmin admin@example.org

  DocumentRoot /var/www

  <Proxy ajp://localhost:8009>
    Allow from all
  </Proxy>

  ProxyPass /idp ajp://localhost:8009/idp retry=5
  ProxyPassReverse /idp ajp://localhost:8009/idp retry=5

  SSLEngine On
  SSLCipherSuite HIGH:MEDIUM:!aNULL:!MD5
  SSLProtocol all -SSLv2 -SSLv3
  SSLVerifyClient optional_no_ca
  SSLVerifyDepth 10
  SSLCertificateFile /opt/shibboleth-idp/credentials/idp.crt
  SSLCertificateKeyFile /opt/shibboleth-idp/credentials/idp.key

  BrowserMatch "MSIE [2-6]" \
    nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0
  # MSIE 7 and newer should be able to use keepalive
  BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown

</VirtualHost>
```

9) Aggiungere “Listen 8443” alle porte ascoltate da Apache2 “/etc/apache2/ports.conf” e attivare il nuovo **idp-ssl** site con “a2ensite idp-ssl”

10) Prelevare la **Terena Chain**:

- `wget https://ca.garr.it/mgt/Terena-chain.pem -O /opt/shibboleth-idp/certs/Terena-chain.pem`

11) Salvare e riavviare Tomcat:

- `service tomcat7 restart`

12) Installazione dell'IdP conclusa, testiamolo!

Aggiungere al file /etc/hosts la seguente riga:

```
127.0.1.1 idp.example.org idp
```

Aprire 1 finestra del Browser e digitare:
`https://idp.example.org/idp/profile/Status`
e deve darvi OK. ==> IdP funzionante su HTTPS

N.B.: Ogni volta che si cambia WAR in /opt/shibboleth-idp/war facendo il suo Undeploy da

Tomcat Manager o altro, BISOGNA ricordarsi di ricreare l'**idp.xml** dentro a `/etc/tomcat7/Catalina/localhost/` che indica a Tomcat7 di prendere il nuovo WAR.

7 Configurare Shibboleth Identity Provider 2.4.0

1) Acquisire i permessi di ROOT:

- `sudo su -`

2) Modificare "logging.xml":

- `vim /opt/shibboleth/conf/logging.xml`

```
<!-- Logs IdP, but not OpenSAML, messages -->
<logger name="edu.internet2.middleware.shibboleth" level="DEBUG"/>

<!-- Logs OpenSAML, but not IdP, messages -->
<logger name="org.opensaml" level="DEBUG"/>

<!-- Logs LDAP related messages -->
<logger name="edu.vt.middleware.ldap" level="DEBUG"/>

<!-- Logs inbound and outbound protocols messages at DEBUG level-->
<logger name="PROTOCOL_MESSAGE" level="DEBUG" />
```

3) Installare expat (necessario per utilizzare il comando xmlwf per la verifica dei file.xml)

- `apt-get install expat`

4) Modificare "handler.xml":

- `vim $IDP_HOME/conf/handler.xml`
 - Disabilitare il blocco relativo all'endpoint RemoteUser (commentandolo)
 - Abilitare il blocco relativo all'endpoint UsernamePassword (decommentandolo)

5) Modificare il file di configurazione login.config

- `vim $IDP_HOME/conf/login.config`

come segue:

a) **Esempio** di connessione a LDAP senza SSL:

```
edu.vt.middleware.ldap.jaas.LdapLoginModule required
ldapUrl="ldap://ldap.example.it:389"
baseDn="dc=example,dc=it"
bindDn="cn=ldapadmin,dc=example,dc=it"
bindCredential="password_serviceUser"
ssl=false
userFilter="uid={0}"
subtreeSearch="true";
```

b) **Esempio** di connessione LDAP con SSL:

```
edu.vt.middleware.ldap.jaas.LdapLoginModule required
ldapUrl="ldaps://ldap.example.it:636"
baseDn="dc=example,dc=it"
bindDn="cn=ldapadmin,dc=example,dc=it"
bindCredential="password_serviceUser"
ssl="true"
userFilter="uid={0}"
subtreeSearch="true";
```

c) **Esempio** di connessione LDAP con TLS:

```
edu.vt.middleware.ldap.jaas.LdapLoginModule required
ldapUrl="ldap://ldap.example.it:389"
baseDn="dc=example,dc=it"
bindDn="cn=ldapadmin,dc=example,dc=it"
bindCredential="password_serviceUser"
tls="true"
userFilter="uid={0}"
subtreeSearch="true";
```

6) Gestire eduPersonTargetID come tipo StoredID:

- apt-get install mysql-server
- cd /usr/local/src/
- wget -O mysql-connector-java-5.1.25.zip http://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.25.zip/from/http://cdn.mysql.com/
- cp /usr/local/src/mysql-connector-java-5.1.25/mysql-connector-java-5.1.25.zip.jar \$TOMCAT_HOME/common/
- cp /usr/local/src/mysql-connector-java-5.1.25/mysql-connector-java-5.1.25.zip.jar \$IDP_HOME/lib/
- mysql -u root -p /* necessario per accedere come root a mysql */
- mysql> SET NAMES 'utf8';
- mysql> SET CHARACTER SET utf8;
- mysql> CHARSET utf8;
- mysql> CREATE DATABASE IF NOT EXISTS userdb CHARACTER SET=utf8;
per creare il database “**userdb**” di test. Vi restituirà:
" Query OK, 1 row affected (0.00 sec) "

- `mysql> grant all privileges on userdb.* to root@localhost identified by 'yourPassword';`
- `mysql> use userdb; /* Così gli dico di usare il database che abbiamo creato */`
- `mysql> CREATE TABLE shibpid`
 - > (
 - > localEntity TEXT NOT NULL,
 - > peerEntity TEXT NOT NULL,
 - > principalName VARCHAR(255) NOT NULL default '',
 - > localId VARCHAR(255) NOT NULL,
 - > persistentId VARCHAR(36) NOT NULL,
 - > peerProvidedId VARCHAR(255) NULL,
 - > creationDate timestamp NOT NULL default CURRENT_TIMESTAMP on update CURRENT_TIMESTAMP,
 - > deactivationDate timestamp NULL default NULL,
 - > KEY persistentId (persistentId),
 - > KEY persistentId_2 (persistentId, deactivationDate),
 - > KEY localEntity (localEntity(16), peerEntity(16), localId),
 - > KEY localEntity_2 (localEntity(16), peerEntity(16), localId, deactivationDate)
 - >) ENGINE=MyISAM DEFAULT CHARSET=utf8;
- `mysql> use mysql;`
- `mysql> INSERT INTO user (Host, User, Password, Select_priv, Insert_priv, Update_priv, Delete_priv, Create_tmp_table_priv, Lock_tables_priv, Execute_priv) VALUES ('localhost', 'idem', PASSWORD('demo'), 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y');`
- `mysql> FLUSH PRIVILEGES;`
- `mysql> GRANT ALL ON userdb.* TO 'idem'@'localhost' IDENTIFIED BY 'demo';`
- `mysql> FLUSH PRIVILEGES;`
- `mysql> QUIT`

7) Modificare il file `$IDP_HOME/conf/attribute-resolver.xml` come segue:

- a) Decomentare tutti gli `<resolver:AttributeDefinition >` ed effettuare le modifiche sottostanti:

```
<resolver:AttributeDefinition
  xsi:type="ad:SAML2NameID"
  id="eduPersonTargetedID"
  nameIdFormat="urn:oasis:names:tc:SAML:2.0:nameid-
    format:persistent"
```

```

        sourceAttributeID="persistentID">
<resolver:Dependency ref="storedID" />
<resolver:AttributeEncoder
    xsi:type="enc:SAML1XMLObject"
    name="urn:oid:1.3.6.1.4.1.5923.1.1.1.10" />

<resolver:AttributeEncoder
    xsi:type="enc:SAML2XMLObject"
    name="urn:oid:1.3.6.1.4.1.5923.1.1.1.10"
    friendlyName="eduPersonTargetedID" />
</resolver:AttributeDefinition>

```

b) **NON DECOMMENTARE** il seguente frammento:

```

<!-- Do NOT use the version of eduPersonTargetedID defined below unless you understand why it was deprecated and know that this reason does not apply to you.
<!--
<resolver:AttributeDefinition xsi:type="ad:Scoped" idID.old" scope="example.it"
sourceAttributeID="persistentID">
<resolver:Dependency ref="storedID" />
<resolver:AttributeEncoder xsi:type="enc:SAML1ScopedString"
name="urn:mace:dir:attribute-def:eduPersonTargetedID" />
</resolver:AttributeDefinition>
-->

```

c) Personalizzare i `<resolver:DataConnector ... />` in modo che:

```

<!-- Example LDAP Connector -->
<resolver:DataConnector id="myLDAP"
    xsi:type="LDAPDirectory"
    ldapURL="ldap://ramo1.di.ldap.da.reperire
            ldap://ramo2.di.ldap.da.reperire
            ldap://ramoN.di.ldap.da.reperire"
    baseDN="dc=dc_di_ldap,dc=dc_di_ldap"
    useStartTLS="true" /* se LDAP con TLS */

    /* Parametri omissibili */
    principal="cn=admin_di_ldap,dc=garr,dc=it"
    principalCredential="password_principal">
/*****

```

```

        <dc:FilterTemplate>
            <![CDATA[
                (uid=$requestContext.principalName)
            ]]>
        </dc:FilterTemplate>
    </resolver:DataConnector>
<resolver:DataConnector xsi:type="StoredId"
    xmlns="urn:mace:shibboleth:2.0:resolver:dc"
    id="storedID"
    sourceAttributeID="uid"
    generatedAttributeID="persistentID"
    salt="Stringa-casuale-generabile-con `openssl rand -base64
36 2>/dev/null` ">
    <resolver:Dependency ref="myLDAP" />

    <ApplicationManagedConnection
        jdbcDriver="com.mysql.jdbc.Driver"
        jdbcURL="jdbc:mysql://localhost:3306/userdb?
autoReconnect=true"
        jdbcUserName="idem"
        jdbcPassword="demo" />
</resolver:DataConnector>

```

- d) Modificare il file `attribute-filter.xml` (quello gestito da IDEM lo potete prelevare da [QUI](#)) per fare in modo che l'IdP rilasci qualche attributo, per esempio:

```

<!-- Release the transient ID to anyone -->
<afp:AttributeFilterPolicy id="releaseTransientIdToAnyone">
    <afp:PolicyRequirementRule xsi:type="basic:ANY"/>
    <afp:AttributeRule attributeID="transientId">
        <afp:PermitValueRule xsi:type="basic:ANY"/>
    </afp:AttributeRule>
</afp:AttributeFilterPolicy>

<afp:AttributeFilterPolicy id="attributesToAnyone">
    <afp:PolicyRequirementRule xsi:type="basic:ANY"/>

    <afp:AttributeRule attributeID="eduPersonTargetedID">
        <afp:PermitValueRule xsi:type="basic:ANY" />
    </afp:AttributeRule>

    <afp:AttributeRule attributeID="eduPersonScopedAffiliation">
        <afp:PermitValueRule xsi:type="basic:ANY"/>
    </afp:AttributeRule>
</afp:AttributeFilterPolicy>

<!-- Specified SP Example -->
<afp:AttributeFilterPolicy id="specifiedSPexample">
    <afp:PolicyRequirementRule
        xsi:type="basic:AttributeRequesterString"
        value="https://entityid.of.sp/shibboleth" />

```

```

<afp:AttributeRule attributeID="commonName">
  <afp:PermitValueRule xsi:type="basic:ANY" />
</afp:AttributeRule>

<afp:AttributeRule attributeID="mail">
  <afp:PermitValueRule xsi:type="basic:ANY" />
</afp:AttributeRule>
</afp:AttributeFilterPolicy>

```

8) Aggiungere le informazioni indicate dal [template di IDEM](#) al metadata del proprio IdP:

- vim /opt/shibboleth-idp/metadata/idp-metadata.xml

9) Prelevare il certificato con cui verranno validati i metadati scaricati:

- cd \$IDP_HOME/credentials
- wget https://idem.garr.it/documenti/doc_download/321-idem-metadata-signer-2019 -O idem_signer_2019.pem

10) Modificare il proprio **relying_party.xml** come segue:

- vim \$IDP_HOME/conf/relying_party.xml

```

<metadata:MetadataProvider id="IDEM-Test-Federation"
  xsi:type="metadata:FileBackedHTTPMetadataProvider"
  metadataURL="http://www.garr.it/idem-metadata/idem-test-metadata-sha256.xml"
  backingFile="/opt/shibboleth-idp/metadata/signed-test-metadata.xml"
  minRefreshDelay="PT5M" maxRefreshDelay="PT4H">

<metadata:MetadataFilter xsi:type="metadata:ChainingFilter">
  <metadata:MetadataFilter xsi:type="metadata:SignatureValidation"
    trustEngineRef="shibboleth.MetadataTrustEngine"
    requireSignedMetadata="true" />
  </metadata:MetadataFilter>
</metadata:MetadataProvider>

```

. . .

(Commentare lo shibboleth.MetadataTrustEngine d'esempio)

```

<!-- Trust engine per la federazione IDEM -->
<security:TrustEngine id="shibboleth.MetadataTrustEngine"
  xsi:type="security:StaticPKIXSignature">
  <security:ValidationInfo id="IDEMFederationCredentials"
    xsi:type="security:PKIXFilesystem">
    <security:Certificate>
      /opt/shibboleth-idp/credentials/idem_signer_2019.pem
    </security:Certificate>
  </security:ValidationInfo>
</security:TrustEngine>

```

11) Riavviare Tomcat:

- service tomcat7 restart

- 12) Registrare i metadati dell'IdP, ottenibili dalla URL
“<https://##idp.example.org##/idp/profile/Metadata/SAML>”
nell' IDEM Entity Registry: <https://registry.idem.garr.it>
(in caso di problemi contattare idem-help@garr.it)

- 13) Verificare che compaia la pagina di Login dopo essere acceduti al proprio IDP dalla pagina
del Service Provider di Test (<https://sp-test.garr.it>) inviato da idem-help@garr.it