Installation script for CzechIdM

LINK to the script *#TODO soon*

About script

The script was developed to make it easier for developers to prepare server and install CzechIdM. This script replaces tutorials Server preparation and Install CzechIdM.

Make sure that your server is running on CentOS. Otherwise, you need to use a different tutorial.

How to use this script

Step 1: Change tomcat mirror in script

In the script find the line, where tomcat is downloaded (only **wget** used there) and change it according to this link.

Step 2: Copy script to the server

Connect to your server and then copy your script there.

```
scp czechidm-install.sh <destination>
```

Step 3: Right to execute

We will need to execute this script, so we need our script to be executable:

```
chmod +x czechidm-install.sh
```

Step 4: Configure YUM repository

Before we start actual script it is advised to change the file CentOS-Base.repo.

As root, edit your /etc/yum.repos.d/CentOS-Base.repo file, to the sections [base] and [updates] append a line:

```
exclude=postgresql*
```

The script will ask to do so in right after it starts. You can just skip it.

Step 5: Start the script

The only thing left is to start the actual script and follow the steps.

Settings after script

Step 1: Apache Tomcat configuration

- Do not show application server version:
  - In the file `/opt/tomcat/current/conf/web.xml` set showServerInfo to false (default is true):

```
    <servlet>
        <servlet-name>default</servlet-name>
        <servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-
class>
        <init-param>
            <param-name>debug</param-name>
            <param-value>0</param-value>
        </init-param>
        <init-param>
            <param-name>listings</param-name>
            <param-value>false</param-value>
        </init-param>
        <init-param>
            <param-name>showServerInfo</param-name>
            <param-value>false</param-value>
        </init-param>
        <load-on-startup>1</load-on-startup>
    </servlet>
```

Step 2: mod_security configuration

Mod_security files locations (on CentOS7):

- Audit log: `/var/log/httpd/modsec\_audit.log`
- Directory with activated rules: `/etc/httpd/modsecurity.d/activated\_rules/`
- basic configuration file for mod_security: `/etc/httpd/modsecurity.d/modsecurity\_crs\_10\_config.conf`
- The file for chosen rules deactivation: `/etc/httpd/conf.d/ssl.conf`

The default set of rules is relatively strict. CzechIdM cannot run with the default configuration of mod_security.

Each rule is identified by a unique ID. If you want to deactivate the whole rule, it is advised to write the rule ID into ssl.conf like this:

```
  <IfModule mod_security2.c>
    SecRuleRemoveById RULE_ID
  </IfModule>
```

Sep 3: mod_security configuration

In the file /etc/httpd/modsecurity.d/modsecurity_crs_10_config.conf, find the rule with id=900012 and add support for content_type=application/json, application/hal+json and text/plain on the line starting with tx.allowed_request_content_type, then allow PUT DELETE and PATCH methods on the line with tx.allowed_methods. Whole rule after the changes looks like this:

```
SecAction \
  "id:'900012', \
  phase:1, \
  t:none, \
  setvar:'tx.allowed_methods=GET HEAD POST OPTIONS PUT PATCH DELETE', \
setvar:'tx.allowed_request_content_type=application/hal+json|application/jso
n|text/plain|application/x-www-form-urlencoded|multipart/form-
data|text/xml|application/xml|application/x-amf', \
  setvar:'tx.allowed_http_versions=HTTP/0.9 HTTP/1.0 HTTP/1.1', \
  setvar:'tx.restricted_extensions=.asa/ .asax/ .ascx/ .axd/ .backup/ .bak/
.bat/ .cdx/ .cer/ .cfg/ .cmd/ .com/ .config/ .conf/ .cs/ .csproj/ .csr/
.dat/ .db/ .dbf/ .dll/ .dos/ .htr/ .htw/ .ida/ .idc/ .idq/ .inc/ .ini/ .key/
.licx/ .lnk/ .log/ .mdb/ .old/ .pass/ .pdb/ .pol/ .printer/ .pwd/
.resources/ .resx/ .sql/ .sys/ .vb/ .vbs/ .vbproj/ .vsdisco/ .webinfo/ .xsd/
.xsx/', \
  setvar:'tx.restricted_headers=/Proxy-Connection/ /Lock-Token/ /Content-
Range/ /Translate/ /via/ /if/', \
  nolog, \
  pass"
```

Step 4: Disabling mod_security rules

In the file `/etc/httpd/conf.d/ssl.conf` deactivate following rules and set their logging:

```
<IfModule mod_security2.c>
        SecRuleRemoveById 981173
        SecRuleRemoveById 960015
        SecRuleRemoveById 950109

        # Allow Czech signs
        SecRuleRemoveById 981318
        SecRuleRemoveById 981242
        SecRuleRemoveById 960024
        SecRuleRemoveById 981245

        # Too restrictive for login format
        SecRuleRemoveById 960035
```

```
        # Needed by Websockets
        <Location "/idm/api/v1/websocket-info/">
                SecRuleRemoveById 970901
        </Location>

        # These break Certificate Authority module
    <Location "/idm/api/v1/crt/certificates/action/validate">
        SecRuleRemoveById 960915
        SecRuleRemoveById 200003
    </Location>

        # do not log request/response body
        SecAuditLogParts ABFHZ
</IfModule>
```

Step 5: mod_deflate configuration

It is advised to set up gzip so the users get minimum of data from the frontend server. In the file `/etc/httpd/conf.d/ssl.conf` we add following configuration and restart the server:

```
<IfModule mod_deflate.c>
        # Compress HTML, CSS, JavaScript, Text, XML and fonts
        AddOutputFilterByType DEFLATE application/javascript
        AddOutputFilterByType DEFLATE application/rss+xml
        AddOutputFilterByType DEFLATE application/vnd.ms-fontobject
        AddOutputFilterByType DEFLATE application/x-font
        AddOutputFilterByType DEFLATE application/x-font-opentype
        AddOutputFilterByType DEFLATE application/x-font-otf
        AddOutputFilterByType DEFLATE application/x-font-truetype
        AddOutputFilterByType DEFLATE application/x-font-ttf
        AddOutputFilterByType DEFLATE application/x-javascript
        AddOutputFilterByType DEFLATE application/xhtml+xml
        AddOutputFilterByType DEFLATE application/xml
        AddOutputFilterByType DEFLATE font/opentype
        AddOutputFilterByType DEFLATE font/otf
        AddOutputFilterByType DEFLATE font/ttf
        AddOutputFilterByType DEFLATE image/svg+xml
        AddOutputFilterByType DEFLATE image/x-icon
        AddOutputFilterByType DEFLATE text/css
        AddOutputFilterByType DEFLATE text/html
        AddOutputFilterByType DEFLATE text/javascript
        AddOutputFilterByType DEFLATE text/plain
        AddOutputFilterByType DEFLATE text/xml
        AddOutputFilterByType DEFLATE application/json
        AddOutputFilterByType DEFLATE application/hal+json

        # Remove browser bugs (only needed for really old browsers)
        BrowserMatch ^Mozilla/4 gzip-only-text/html
        BrowserMatch ^Mozilla/4\.0[678] no-gzip
```

```
        BrowserMatch \bMSIE !no-gzip !gzip-only-text/html
        Header append Vary User-Agent
</IfModule>
```

Step 6: Application properties

* The most important file is **/opt/czechidm/etc/application-production.properties** (application-PROFILE.properties, where the PROFILE is the profile you run the IdM under). You can use most of the file as-is, there is a bit of configuration needed though. This is a template file:

application-production.properties

```
# Doc: https://wiki.czechidm.com/devel/dev/configuration/backend

idm.pub.app.instanceId=idm-primary
idm.pub.app.stage=production

spring.datasource.url=jdbc:postgresql://localhost:5432/czechidm
spring.datasource.username=czechidm
spring.datasource.password=********** TODO *********
spring.datasource.driver-class-name=org.postgresql.Driver
spring.datasource.tomcat.validationQuery=SELECT 1
spring.datasource.tomcat.test-on-borrow=true
spring.jpa.generate-ddl=false
spring.jpa.hibernate.ddl-auto=none
flyway.enabled=true

scheduler.enabled=true
scheduler.task.queue.process=1000
scheduler.properties.location=quartz-production.properties
logging.config=/opt/czechidm/etc/logback-spring.xml
idm.sec.core.demo.data.enabled=false

#spring.cache.ehcache.config=classpath:ehcache.xml

spring.activiti.processDefinitionLocationPrefix=classpath*:/eu/bcvsolut
ions/idm/workflow/
idm.sec.core.notification.template.folder=classpath*:/eu/bcvsolutions/i
dm/templates/
idm.sec.core.script.folder=classpath*:/eu/bcvsolutions/idm/scripts/
# configuration property for default backup
idm.sec.core.backups.default.folder.path=/opt/czechidm/backup


idm.pub.security.allowed-origins=http://localhost
# Generate JWT token security string as "cat /dev/urandom | tr -dc 'a-
z0-9' | head -c VALUE" where VALUE can be from 1 to 255.
# We recommend the VALUE to be at least 25.
idm.sec.security.jwt.secret.token=********** TODO *********
idm.sec.security.jwt.expirationTimeout=36000000
```

```
# recaptcha
# - recaptchaservice endpoint
#idm.sec.security.recaptcha.url=https://www.google.com/recaptcha/api/si
teverify
# - secret key, can be generated here
https://www.google.com/recaptcha/admin
idm.sec.security.recaptcha.secretKey=xxx
# Proxy for HTTP requests
#idm.sec.core.http.proxy=12.34.56.78:1234

# Cipher secret key for crypt values in confidential storage
# for crypt values is used secretKey or secretKey defined by file -
secretKeyPath
#cipher.crypt.secret.key=XXXXXXXXXXXXXXX
cipher.crypt.secret.keyPath=/opt/czechidm/etc/secret.key


idm.sec.core.emailer.test.enabled=true
# http://camel.apache.org/mail.html
idm.sec.core.emailer.protocol=smtp
idm.sec.core.emailer.host=something.tld
idm.sec.core.emailer.port=25
# idm.sec.core.emailer.username=czechidm@domain.tld
# idm.sec.core.emailer.password=password
idm.sec.core.emailer.from=czechidm@localhost

## Global property that allow disable or enable sending notification
from WF
idm.sec.core.wf.notification.send=false


# supports delete identity
idm.pub.core.identity.delete=true
#
# default password change type for custom users, one of values:
# DISABLED - password change is disable
# ALL_ONLY - users can change passwords only for all accounts
# CUSTOM - users can choose for which accounts change password
idm.pub.core.identity.passwordChange=ALL_ONLY
#
# required old password for change password
idm.pub.core.identity.passwordChange.requireOldPassword=true
#
# create default identity's contract, when identity is created
idm.pub.core.identity.create.defaultContract.enabled=true


# Default user role will be added automatically, after an identity is
logged in
# could contains default authorities and authority policies
```

```
configuration
# for adding autocomplete or all record read permission etc.
idm.sec.core.role.default=userRole
# Admin user role
idm.sec.core.role.admin=superAdminRole


# ID system against which to authenticate
idm.sec.security.auth.systemId=

# attachments will be stored under this path.
# new directories for attachment will be created in this folder
(permissions has to be added)
# System.getProperty("user.home")/idm_data will be used if no path is
given
idm.sec.core.attachment.storagePath=/opt/czechidm/data
```

Adjust database configuration

If you followed this howto, the only thing you should need to adjust is a **spring.datasource.password** propetry. Set it to the password for czechidm user in PostgreSQL. If necessary, adjust other database connection properties...

```
spring.datasource.url=jdbc:postgresql://localhost:5432/czechidm
spring.datasource.username=czechidm
spring.datasource.password=********** TODO *********
spring.datasource.driver-class-name=org.postgresql.Driver
spring.datasource.tomcat.validationQuery=SELECT 1
spring.datasource.tomcat.test-on-borrow=true
```

Generate JWT token

Set value of the **idm.sec.security.jwt.secret.token** property as is described in the template file:

```
# Generate JWT token security string as "cat /dev/urandom | tr -dc 'a-z0-9'
| head -c VALUE" where VALUE can be from 1 to 255.
# We recommend the VALUE to be at least 25.
idm.sec.security.jwt.secret.token=********** TODO *********
```

Local confidential storage

Local confidential storage is encrypted by AES algoritm. Read more. Confidential storage is encrypted by a key found in **secret.key** file you already created.

There are two properties in application-production.properties that influence the confidential storage:

- You can set the 128bit (16byte) key directly in the property file using **cipher.crypt.secret.key** property or
- you can create separate file (in our case **secret.key**) containing a random string. Then you reference this file with **cipher.crypt.secret.keyPath** property.



CzechIdM doesn't contain any default key for crypt confidential storage. Please define it before you start using the IdM.

Confidential storage uses AES/CBC/PKCS5Padding (more info) algorithm which operates with 128bit key.

Attachment store

In CzechIdM, users can sometimes add attachments (say, attach *.jpeg photo to their employee card request). Those files are stored in the attachment store. With the following property, you can configure, where the store is. If you used sample property file, the store is by-default located under /opt/czechidm/data .

```
# attachments will be stored under this path.
# new directories for attachment will be created in this folder (permissions has to be added)
# System.getProperty("user.home")/idm_data will be used if no path is given
idm.sec.core.attachment.storagePath=/opt/czechidm/data
```

Step 7: Allow network services

Firewall may restrict the access to all port except ssh (22/tcp). To be able to use CzechIdM, allow port 443/tcp and reload firewalld:

```
firewall-cmd --permanent --add-port=443/tcp
firewall-cmd --reload
```

From:
https://wiki.czechidm.com/ - **CzechIdM Identity Manager**

Permanent link:
https://wiki.czechidm.com/tutorial/adm/installation_script_-_centos

Last update: **2019/04/12 10:39**