Abstract

This text briefly described how to make XAdmin reuse some pre-installed version of (either of) Apache, PHP, OpenSSL, and MariaDB instead of using XAdmin's own versions.

A certain amount of Linux experience is needed for understanding & applying the contained info.

In general, XAdmin is tested against its own versions of these tools. Thus, if you aim for best *compatibility* then you should rather stick to these versions. However, as XAdmin-releases are built rather rarely, you may sometimes prefer to use some different/newer versions (at least until next XAdmin-release) in order to get some *latest security patches* quickly.

Most PHP-scripts of XAdmin are called directly by Apache (via PHP *module*) but some (e.g. XAdmin-internal daemons) are called by shell instead (via PHP *binary*). It is possible to use *different* PHP-versions for these two cases but, for best compatibility, we recommend that you use either only built-in Apache & PHP or only host-provided Apache & PHP.

In context of XAdmin, OpenSSL is called by Apache & PHP (via SSL *library*) but also by shell (via SSL *binary*). Like in previous remark, for best compatibility, we recommend not to mix versions.

Table of Contents

1. Host system setup

For our tests, we had used ISO-file “CentOS-7-x86_64-Everything-2009.iso”.

For SW-setup, we had chosen “simple webserver” with following additions. However, other combinations should generally also work.

- Guest agents
- MariaDB client
- PHP support
- Perl for web
- Remote administration
- Compatibility libraries
- Development tools
- System-admin tools

2. Firewall settings

After general network-setup, we had set the firewall to allow HTTP/HTTPS-access.

```
firewall-cmd --zone=public --permanent --add-service=http -add-service=https
```

You may also optionally allow TFTP if using it for file-transfers instead of the HTTPS default.

```
firewall-cmd --zone=public --permanent --add-service=tftp
```

3. DB-server installation & settings

Following steps are *not* needed if you are using XAdmin’s built-in DB-server.

For our tests, we had installed & started the MariaDB server.

```
yum install mariadb-server
systemctl restart mariadb
```

To prepare the DB-server for access by XAdmin, we used the following commands.

- `mysql_secure_installation`

  Specify root password and confirm proposed security-settings.

- `mysql -u root -p`
  - Enter root-pwd to access mysql-shell, then type following command.

    ```
    CREATE DATABASE confix;
    ```

  - Optionally, if you want XAdmin to use a different DB-user than the default-user `root`, type following commands in mysql-shell (replace text in *red italics* by real data).

    ```
    CREATE DATABASE confix;
    CREATE USER 'youruser'@'localhost' IDENTIFIED BY 'yourpassword';
    ```

    (make sure to use same user/password here as in XAdmin-installer)

    ```
    GRANT ALL PRIVILEGES ON confix.*  TO 'youruser'@'localhost';
    FLUSH PRIVILEGES;
    ```

  - Leave mysql-shell by entering the ‘`quit`’ command

4. PHP update to version 7.4 & settings

Following steps are *not* needed if you are using XAdmin’s built-in PHP & Apache.

Remark: The PHP-scripts of XAdmin have generally already been tested against PHP 8.1. However, the needed IONCUBE-loader does not yet support any versions newer than 7.4.

The regular PHP-version of CentOS 7 is rather old. For upgrading PHP to 7.4, we had followed the steps of the configuration wizard of Remi's RPM repository.

https://rpms.remirepo.net/wizard/

This covers already most PHP-modules needed by XAdmin. In addition, we had installed the following ones afterwards.

```
yum install php-mysqlnd
yum install php-mbstring
yum install php-bcmath
```

In our tests, we had to reboot the host afterwards to apply the changes.

To see what modules are currently installed, you can use e.g. the following command.

```
php -m
```

For proper support of XAdmin’s PHP-scripts, we had to change/add the following lines (each at a different place) in the PHP-settings in `/etc/php.ini`.

- `zend_extension = "/xadmin/lib/ioncube_loader.so"`
- `default_charset = "iso-8859-1"`
- `include_path = "/xadmin/include/xadmin:."`
- `date.timezone = 'Europe/Berlin'`

The mentioned file `ioncube_loader.so` is installed together with XAdmin (see next section). It has been extracted (and renamed) from the official archive that can be downloaded from the following location.

https://www.ioncube.com/loaders.php

If you use a non-default Apache/PHP-installation then you may need the “thread-safe” variant of this dynamic library instead.

5. XAdmin installation

5.1. Run installer script

DB-server and PHP should be ready by now. So, you may now run XAdmin's installer-script unless you already have an installed XAdmin. In the latter case, please proceed to the next section instead.

For each of the mentioned 3rd party tools, the script will ask you if you prefer your own version.

- For own *DB-server*, specify its `<host>:<port>:<socket>` address. In case of local MariaDB-server, it is usually the following one.

  ```
  localhost:3306:/var/lib/mysql/mysql.sock
  ```

- For own *PHP-binary*, specify its path, usually the following one.

  ```
  /usr/bin/php
  ```

- For own *SSL-binary*, specify its path, usually the following one.

  ```
  /usr/bin/openssl
  ```

- For own *Apache* (and related PHP/SSL-library), answer with “`n`” when script asks you whether to use XAdmin's builtin version.

After installation, you may already start XAdmin e.g. with the following command.

```
/xadmin/script/xadminctl restart
```

In case that you configured XAdmin to run with an “external” HTTP-server, you will now have to adjust some *Apache-settings* to get the web-service running, as described in a later section.

5.2. Optionally, adjust control-scripts

When XAdmin is configured to run with “remote” HTTP/DB-servers then, by default, the related *control-scripts* `/xadmin/scripts/apachectl` or `/xadmin/scripts/mysqlctl` will *not* start/stop the related service (mainly because not “feeling responsible” for doing so). You may change this default behaviour by adjusting the mentioned two scripts accordingly. This has already been prepared but yet “disabled” for the case of `httpd` and `mariadb` being controlled by `systemd`. Just search these scripts for “`Disable lines above`” (two findings per file) and follow instructions there.

6. Decoupling an already installed XAdmin

Even with an already installed XAdmin (of at least version 4.2.1), you can switch to using your own versions of the mentioned tools by following the steps below. All other sections of this document except for the previous one apply to both cases (i.e. new or already installed XAdmin).

6.1. Stop XAdmin

To avoid any data-corruption, make sure that XAdmin is no longer running.

```
/xadmin/scripts/xadminctl stop
```

6.2. Backup XAdmin directory

This step is optional but highly recommended. For a typical XAdmin installation, directory `/xadmin` contains all relevant recovery-info (including DB-contents) in case of any unexpected problems.

```
tar czf xadmin-backup.tar.gz /xadmin
```

Remark: In case that you switch to a different DB-server, the DB-contents will later be stored at a different location. So, for later backups, you would have to consider also this further directory.

6.3. Take snapshot of old DB-server

This step is relevant only if you want to switch the DB-server.

```
/xadmin/scripts/mysqlctl start
```

```
/xadmin/bin/mysqldump -u root -p -c -q -B confix --add-drop-table > confix.sql
```

(then enter password for root DB-user)

```
/xadmin/scripts/mysqlctl stop
```

6.4. Change XAdmin-settings

- For different *DB-server*
  - File `/xadmin/scripts/environment`
    - (Var `XADMIN_DBHOST`: Keep host at value of "`localhost`".)
    - Var `XADMIN_DBPORT`: Set new DB-port, e.g. "`3306`".
    - Var `XADMIN_DBSOCK`: Set new DB-socket-path, e.g. "`/var/lib/mysql/mysql.sock`".
    - Var `XADMIN_DBREMOTE`: Change `0` to `1`.
  - File `/xadmin/etc/confix.xml`
    - Field `<database><server>`: Change to new server-address (*`host`*`:`*`port`*`:`*`path`*).
    - Field `<database><dbremote>`: Change `0` to `1`.
  - File `/xadmin/var/mysql/my.cnf`
    - Var `port` (appears *twice*): Set new DB-port, e.g. "`3306`".
    - Var `socket` (appears *twice*): Set new socket-path, e.g. "`/var/lib/mysql/mysql.sock`".
- For different *PHP-binary*
  - File `/xadmin/scripts/environment`, var `PHP`: Set new command, e.g. "`/usr/bin/php -q`".
  - File `/xadmin/etc/confix.xml`
    - Field `<scanner><methods><config><pgmcall>`: Change part of PHP-binary-path.
    - Field `<scanner><methods><config><extcall>`: Change part of PHP-binary-path.
    - Field `<config><php><bin>`: Set new binary-path, e.g. "`/usr/bin/php`".
- For different *SSL-binary*
  - File `/xadmin/scripts/environment`, var `SSLBIN`: Set new path, e.g. "`/usr/bin/openssl`".
  - File `/xadmin/etc/confix.xml`, field `<config><ssl><bin>`: Set new path.
- For different *Apache*
  - File `/xadmin/scripts/environment`, var `XADMIN_HTTPREMOTE`: Change `0` to `1`.
  - File `/xadmin/etc/confix.xml`, field `<config><http><httpremote>`: Change `0` to `1`.

6.5. Replay snapshot on new DB-server

This step is relevant only if you want to switch the DB-server. The commands listed below refer to the case of using `mariadb` with `systemd` as the new DB-server.

```
systemctl start mariadb
mysql -p < confix.sql
```

6.6 Restart XAdmin

Now, you can restart XAdmin again.

```
/xadmin/scripts/xadminctl start
```

In case that you configured XAdmin to run with an "external" HTTP-server, you still have to adjust some *Apache-settings* to get the web-service running, as described in the next section.

Furthermore, as described in the previous section, you may optionally tweak the control-scripts for starting/stopping also any "external" HTTP/DB-server.

7. Apache settings

Following steps are *not* needed if you are using XAdmin's built-in Apache.

For a pre-installed Apache to properly handle the web-contents of XAdmin, some parts of its configuration must be adjusted. Here is a summary of the changes done for our internal tests with a default-configured Apache.

7.1 Disable Apache default page

Disable all lines in `/etc/httpd/conf.d/welcome.conf` by adding a leading "`#`".

7.2. Change httpd user

Adjust following lines in `/etc/httpd/conf/httpd.conf`.

```
User confix
Group confix
```

7.3 Prepare access to XAdmin web-content

Adjust following lines in `/etc/httpd/conf/httpd.conf` …

```
DocumentRoot "/xadmin/htdocs"
```

… and …

```
<Directory "/xadmin/htdocs">
    AllowOverride None
    Require all granted
</Directory>
```

Furthermore, for *SELinux* to allow Apache to access related files, enter following two commands (for some background-info, see e.g. linux.die.net/man/8/httpd_selinux).

```
chcon -R --type=httpd_sys_rw_content_t /xadmin
chcon -R --type=bin_t /xadmin/bin /xadmin/lib
```

To check if SELinux is enabled on your system, you can use e.g. the following command.

```
selinuxenabled 2> /dev/null && echo "ENABLED" || echo "DISABLED"
```

Remark: For convenience, "`/xadmin/scripts/apachectl caps`" also runs these commands.

7.4 Apply self-signed certificate

Adjust following lines (each at a different place) in `/etc/httpd/conf.d/ssl.conf`.

- `SSLCertificateFile /xadmin/etc/xadmin.crt`
- `SSLCertificateKeyFile /xadmin/etc/xadmin.key`

This self-signed certificate is built by "`/xadmin/scripts/apachectl cert`". You may prefer to get/install a proper one e.g. from letsencrypt.org but the details are beyond the scope of this doc.

7.5 Enable file-transfer (of firmware, certs, or configs)

Add following lines in `/etc/httpd/conf/httpd.conf` …

```
DavLockDB /xadmin/var/run/DavLock
```

… and …

```
<Location /tftpboot>
    Dav On
    Require valid-user
    AuthType Basic
    AuthName DAV
    AuthUserFile /xadmin/etc/httpd.passwd
</Location>
```

The referenced file `httpd.passwd` is built by "`/xadmin/scripts/apachectl passwd`", based on the values of fields `<config><fileTransferUsr>/<fileTransferPwd>` in file `/xadmin/etc/confix.xml`. Please do not forget to re-run this command whenever you change these fields.

7.6 Make log-files available in XAdmin web-interface

XAdmin's web-interface expects these logs to lie in directory `/xadmin/var/log`. To make them available there, change following line in `/etc/httpd/conf/httpd.conf` …

```
ErrorLog "/xadmin/var/log/http_error.log"
```

… and following one in `/etc/httpd/conf.d/ssl.conf`.

```
ErrorLog /xadmin/var/log/https_error.log
```

7.7 Enable XAdmin-specific environment-variables

The PHP-scripts to be run by Apache expect the following environment-variables to be set.

```
XADMIN_BIN, XADMIN_CONFIG, XADMIN_ROOT, XADMIN_RUN
```

These are specified (beside others) in file `/xadmin/scripts/environment`. Unfortunately, there seems to be no easy/clean way to make `systemd` include this file when starting `httpd`. Thus, for our tests, we had added the following lines to the end of file `/etc/sysconfig/httpd` instead.

```
XADMIN_BIN="/xadmin/bin"
XADMIN_CONFIG="/xadmin/etc"
XADMIN_ROOT="/xadmin/htdocs"
XADMIN_RUN="/xadmin/var/run"
```

7.8 Restart Apache

Can be done e.g. with the following command.

```
systemctl restart httpd
```