Adobe Commerce – Cybersource Installation guide

September 2022

cybersource
A Visa Solution

Table of Contents

1. Magento Installation

1.1. Prerequisites

Ensure following are installed:

- PHP
- MySQL
- Composer
- Elasticsearch 7
- In the php.ini file, find these rows and remove " ; " before the below mentioned row:

  ;extension=intl,
  ;extension=soap,
  ;extension=sockets,
  ;extension=sodium,
  ;extension=xsl,

1.2. Install Magento through composer

Run the following command in Magento Root Directory :

composer create-project --repository-url=https://repo.magento.com/ magento/project-community-edition magento245

cd magento245

composer install

1.3. File Correction

a) Correct File Gd2.php [\vendor\magento\framework\Image\Adapter\Gd2.php]

Replace below line of code:

```
if ($url && isset($url['scheme']) && !in_array($url['scheme'], $allowed_schemes)) {
```

with

```
if ($url && isset($url['scheme']) && !in_array($url['scheme'], $allowed_schemes) && !file_exists($filename)) {
```

b) Correct File Validator.php [\vendor\magento\framework\view\Element\Template\File\Validator.php]

Insert below line of code:

```
$realPath = str_replace('\\', '/', $realPath);
```

After below line of code:

```
$realPath = $this->fileDriver->getRealPath($path);
```

c) Correct File PluginListGenerator.php [\vendor\magento\framework\Interception\PluginListGenerator.php]

Replace below line of code:

```
$cacheId = implode('|', $this->scopePriorityScheme) . "|" . $this->cacheId;
```

With

```
$cacheId = implode('-', $this->scopePriorityScheme) . "-" . $this->cacheId;
```

d) In app\etc\di.xml, replace Symlink with Copy

```
<virtualType name="developerMaterialization"
type="Magento\Framework\App\View\Asset\MaterializationStrategy
\Factory">
      <arguments>
            <argument name="strategiesList" xsi:type="array">
                  <item name="view_preprocessed"
xsi:type="object">Magento\Framework\App\View\Asset\Materializa
tionStrategy\Symlink</item>
                  <item name="default"
xsi:type="object">Magento\Framework\App\View\Asset\Materializa
tionStrategy\Copy</item>
            </argument>
      </arguments>
</virtualType>
```

1.4. Magento Installation

Run the below command in Magento Root Directory:

```
php bin/magento setup:install --base-url-secure="url" --db-
host="host" --dbname="dbname" --db-user="dbuser" --db-
password="dbuserpassword" --adminfirstname="fname" --admin-
lastname="lname" --admin-email="email" --adminuser="uname" --
admin-password="password" --use-rewrites="1" --
backendfrontname="admin"
```

Descriptions:

base-url: the path that your Magento directory is in, which follows the following format: http[s]://<host or ip>/<your Magento install dir>/

db-host: the hostname or IP address of your host

db-name: change it to the name of the Magento database you just created

db-user: a database user with full permission. We'll be using the default root user.

db-password: the password of your database user. Leave it blank if you're using 'root' database user

admin-firstname: your first name

admin-lastname: your lastname

admin-email: your email address

admin-user: the username which you'll be using to log into Admin Panel

admin-password: the password which you'll be using to log into Admin Panel.

use-rewrites: set to 1 to enable Web Server Rewrites. This will help with your site ranking.

backend-frontname: set your Admin URL. Omitting this parameter will result in a randomly generated URL for your Magento Admin path (e.g., admin_jkhgdfq)

For more configurable options, please refer to the official guide by Magento.

Upon successful installation, you will see similar message:

```
Post installation file permissions check…
For security, remove write permissions from these directories:
'C:/xampp/htdocs/magento24/app/etc'
[Progress: 1270 / 1270]
[SUCCESS]: Magento installation complete.
[SUCCESS]: Admin Panel URI: /admin
Nothing to import
```

1.5. Run Magento Command

In CMD [Magento root directory], run following commands:

```
php bin/magento module:disable Magento_TwoFactorAuth
php bin/magento setup:di:compile
php bin/magento indexer:reindex
php bin/magento setup:upgrade
php bin/magento setup:static-content:deploy -f
php bin/magento cache:clean
php bin/magento cache:flush
```

Refer to official Magento v2.x extensions installation guide:

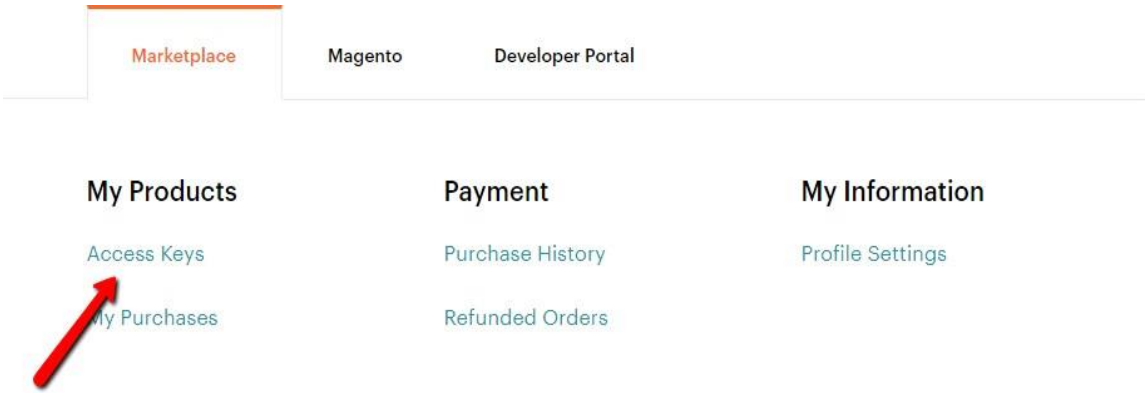
https://devdocs.magento.com/guides/v2.4/install-gde/install-flow diagram.html

2. Install Cybersource Module

We can have Cybersource installation through Marketplace of Magento.

1. Place an order on Magento Marketplace with Cybersource module https://marketplace.magento.com/cybersource-global-payment-management.html

2. Go to My Profile -> My Products -> Access Keys and copy keys



3. Rename auth.json.sample file to auth.json in magento root directory on your server and enter the access keys from step 2.

```
{
  "http-basic": {
    "repo.magento.com": {
      "username": "Your_Public_Key",
      "password": "Your_Private_key"
      }
  }
}
```

4. Open composer.json and add the following keys and values under the "require" array

```
        "version": "2.4.5",
            "require": {
                "goetas-webservices/xsd2php-runtime": "^0.2.15",
                "lcobucci/jwt": "4.1.5",
    }
```

5. Run commands on your magento root directory

composer require cybersource/global-payment-management

php bin/magento module:enable CyberSource_AccountUpdater CyberSource_Address CyberSource_ApplePay CyberSource_Atp CyberSource_BankTransfer CyberSource_Core CyberSource_ECheck CyberSource_KlarnaFinancial CyberSource_PayPal CyberSource_SecureAcceptance CyberSource_Tax CyberSource_VisaCheckout

```
php bin/magento module:disable Magento_TwoFactorAuth
php bin/magento setup:di:compile
php bin/magento indexer:reindex
php bin/magento setup:upgrade
php bin/magento setup:static-content:deploy -f
php bin/magento cache:clean
php bin/magento cache:flush
php bin/magento module:status
```

3. Updating the Cybersource bundle with Latest version

Step 1: Navigate to Magento root directory → composer.json file

Step 2: In composer.json file, under "require" field change the version for our plugin with latest version.

Step 3: After changing the version in "require" field of composer.json, run the composer update command.

4. Appendix A – Security Best Practices

4.1. Securing Files

Make sure your installation files are only accessible locally by properly setting up permissions and .htaccess file. Set up file permissions based on 'need to know' and 'least privilege' and ensure that all files that govern access to parts of the application are secured. Ensure files are not accessible over the web interface.

For more information please refer to:
https://blog.nexcess.net/2010/12/06/securing-magento-file-directory-permissions/

4.2. Password Policy

Enforce strong password requirements to ensure the application is protected from a brute force attack. For more information, please refer to:

NIST Digital Identity Guidelines: https://pages.nist.gov/800-63-3/

4.3. SSL/TLS Encryption

Ensure only the latest TLS standard is enabled on any connections. Explicitly disable any TLS versions that are not current (at time of writing, only TLS 1.2 is not deprecated).

Ensure that Cipher suites that have been deprecated are disabled. For more information, please

refer to:

OpenSSL Cipher Suite Names: https://www.openssl.org/docs/manmaster/man1/ciphers.html -CIPHER-SUITE-NAMES

Apache HTTPS Cipher Suite Restriction:
http://httpd.apache.org/docs/current/ssl/ssl_howto.html

4.4. Software Versions

Ensure all software versions are on the very latest version. Examples are PHP, Java and the Magento software itself. Versions that are branched should have the latest patches from that branch installed. For more information, please refer to:

Magento: https://magento.com/security/patches

PHP: http://php.net/downloads.php

Apache Security Vulnerabilities: http://httpd.apache.org/security_report.html

4.5. Removing Default Web Servers

Software packages such as Apache install default web pages and/or web server instances. If the web server instance is not required, it is best practice to disable the service. If the service is needed, remove default pages and default install directories (/docs, /examples, etc.). For more information, please refer to:

Google Hacking Mini-Guide:
http://www.informit.com/articles/article.asp?p=170880&seqNum=2&rl=1

SecurityFocus - Securing Apache:
http://www.securityfocus.com/infocus/17
86

4.6. Securing Web Servers

The following steps can be taken to reduce and/or eliminate the risk of information disclosure because of using hostnames in URLs:

- Use local domain names rather than IP addresses.
- Remove references to backend system names, IP's, and ports.
- Do not disclose system and/or program user IDs to application users.
- Maintain all error codes and debug information in non-user accessible error logs.

For more information, please refer to:

OWASP -Security by Design Principles:
https://www.owasp.org/index.php/Security_by_Design_Principles

The following steps can be taken to secure insecure commands on Apache:

Use the Apache mod rewrite module to deny HTTP requests or to permit only the methods needed to meet site requirements and policy. Prohibited HTTP methods can be disabled with the following mod_rewrite syntax.

RewriteEngine On RewriteCond %{REQUEST_METHOD}
^TRACE|TRACK|PUT|DELETE|HEAD|OPTIONS|CONNECT

RewriteRule .* - [F]

Alternatively, note that Apache versions 1.3.34, 2.0.55, and 2.2 support disabling the TRACE method natively via the TraceEnable directive.

For more information, please refer to:

Testing for HTTP Methods:
https://www.owasp.org/index.php/Test_HTTP_Methods_(OTG-CONFIG-006)

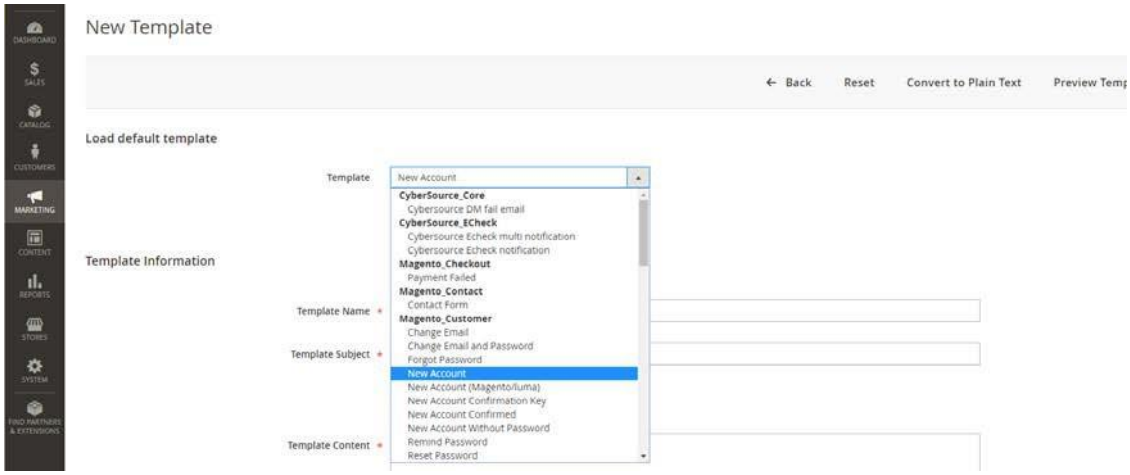
Apache HTTP Server mod_rewrite:
http://httpd.apache.org/docs/mod/mod_rewrite.html

4.7. Password Emails

Magento emails the password to users in plaintext as standard. This is not good security practice and can lead to information disclosure attacks via social engineering or other means (copying). To protect your customers from this, it is recommended to remove plaintext passwords sent via email. The steps to prevent this are as below:

In Magento, do the following:

1. From admin, go to Marketing / Email Templates
2. Click Add New Template orange button
3. In Load default template, select New Account template and load it



4. After load template, in Template Content, find a line show plaintext password and remove it
5. Click Save Template.
6. For more information, please refer to:
7. NIST Digital Identity Guidelines SP 800-63-3: https://pages.nist.gov/800-63-3/

4.8. Secure Authentication Credentials

Authentication Credentials must be stored in a secure manner, according to industry good practice. Securing authentication credentials should be via methods such as strong Encryption, using industry standard encryption methodologies. For more information, please refer to:

National Institute of Standards and Technology: https://pages.nist.gov/800-63-3/

4.9. Security Good Practice

Any implementation of the Magento software package should be undertaken with care. Due diligence should be performed when looking at configuration settings and industry good practice guidelines should be always followed. Cyber Security attacks and subsequent breeches can be brand damaging and put customer's personal data at risk.

For more guidelines on general security good practice, please see the following external sources:

National Institute of Standards and Technology: https://www.nist.gov/

PCI DSS: https://www.pcisecuritystandards.org/pci_security/

Center for Internet Security: https://www.cisecurity.org/

Magento Security Best Practices: https://magento.com/security/best-practices

OWASP: https://www.owasp.org/index.php/Main_Page

SANS Institute: https://www.sans.org/

International Organization for Standardization (ISO) – ISO 27001 and 27002 and any other applicable standards: https://www.iso.org/standards.html