

Getting Commerce-Ready with NuSphere and PHP

By Julie Meloni, Web Applications Developer

TABLE OF CONTENTS

Certificates and
Cryptography

Testing Your Server

Obtaining an SSL
Certificate

Installing Your SSL
Certificate

Restarting Apache,
the Secure Way

Preparing for
Credit Card
Transactions

Using PHP
Functions to
Process
Transactions

Helpful Hints

About NuSphere

Now that you have NuSphere installed on a Linux or Windows server, you may be interested in getting some sort of shopping application up and running. The applications installed by NuSphere lay the groundwork for running a secure server ready for e-commerce. But you will not be able to complete credit-card transactions until you attend to a few details. This paper covers everything you will need to do to get your server configured for secure transactions.

Before you deploy a commerce application on the Internet, you should do everything in your power to make your server safe from intruders. For primers on systems security, make sure to read “Securing Your NuSphere Installation on Red Hat Linux” (http://www.nusphere.com/products/library/secure_install_redhat.pdf) and other helpful documents in the NuSphere Tech Library).

Certificates and Cryptography

Before modifying your installation, it's important to understand what a certificate is, what it does, and a little bit about cryptography. By following the directions in this paper you will produce a server capable of conducting secure transactions, but if you know why you are performing each of these steps, you will have a much easier time keeping your server secure in the future.

A certificate, which you'll be purchasing from a Certification Authority such as Verisign, binds a specific server to a set of *keys* that are used to encrypt information. These keys are part of the system of Public Key Cryptography. Public Key Cryptography assigns two complimentary keys, one public and one private, to an entity. In this case, the entity is your server. When information is encrypted using the private key, only the specific, complimentary public key can be used to decrypt it, and vice versa—any information encrypted using the public key can only be decrypted by the private key.

Consider this example of Public Key Cryptography: Your server has a public key and a private key. Your bank also has a public key and a private key. Your server and the bank want to send encrypted messages to each other, so they exchange public keys. Your server has its own private key as well as the bank's public key. The bank has its own private key as well as your server's public key. When your server wants to send an encrypted message to the bank, it uses encryption software to scramble the message based on the bank's public key. The bank receives the message, then uses its encryption software and its private key to decrypt it. Only the bank will be able to decrypt a message that has been encrypted by someone using its public key.

Your server certificate makes your server (and its keys) act as one end of the connection, and the user's web browser (or a bank) act as the other. When you, as a web surfer, access a secure server, your browser will accept the server's certificate and its public key. The browser will then encrypt all of the data transmitted between itself and the web server based on that key. The web server decrypts the data using its private key and displays the appropriate information.

Testing Your Server

Eventually you will need to get and install a certificate, but before you do that, it's good practice to test your installation. Your NuSphere installation has already handled the most difficult aspects of securing your site: installing `mod_ssl`, preparing the Apache configuration files to handle a secure server, and readying a "dummy" certificate for testing purposes.

On Linux, you normally issue this command to start your Apache server:

```
/usr/local/nusphere/apache/bin/apachectl start
```

This command starts Apache in standard (non-secure) mode. To start Apache in secure mode, simply use:

```
/usr/local/nusphere/apache/bin/apachectl startssl
```

This command tells Apache to enable the SSL module and to listen on the secure port. Normally the https port is 443, and you will want to make sure you are using that port when your site is live so that users will not have to enter a port number in their browsers. By default, NuSphere is configured so that the secure port is 9443. This can be changed by altering the **Listen** element for the SSL configuration in the `httpd.conf` file.

On Windows, the Start Apache item from Start Menu automatically enables Apache with SSL support.

To test your secure server with the dummy certificate, open your browser to `https://[your server]:[secure port]` and, depending on your web browser of choice, you should see several warning messages about how the name on the certificate does not match the name of the server. If you accept the warnings and continue, you will see the secure key in your browser toolbar indicating that all data transferred between your browser and the server is indeed secure.

Now that you have verified the installation of the dummy certificate, and also that your server is prepared to handle secure services, it's time to get a real certificate from one of the Certification Authorities.

Obtaining an SSL Certificate

The recommended Certification Authorities are Verisign (<http://www.verisign.com/>) and Thawte (<http://www.thawte.com/>). Incidentally, Verisign now owns Thawte, as well as other secure services companies, so just bookmark the Verisign web site now.

Before you can begin the process of filling out the form and paying your money to obtain your secure certificate, you must use **OpenSSL** to create an RSA private key for your server, as well as a certificate signing request (CSR). These two pieces are crucial, as the actual SSL certificate is based on the information in both the private key and the CSR.

On Linux, navigate to the `/usr/local/nusphere/ssl/bin` directory and type the following:

```
openssl genrsa -des3 -out server.key 1024
```

You will be asked for a passphrase, and then to verify that passphrase. **DO NOT FORGET IT!** You should also backup the `server.key` file and store it in a safe place.

Windows users, open a DOS console and enter the `C:\Program Files\nusphere\apache\` directory. The OpenSSL binary is found in this directory, and the command is the same.

Next, you'll create the certificate signing request, which the Certificate Authority uses to extract information about your company and produce the SSL certificate. On Linux, in the `/usr/local/nusphere/ssl/bin` directory, type:

```
openssl req -new -key server.key -out server.csr
```

This will start a short interactive script where you will be asked to enter some basic information about you and your company. When this process is complete, you should have a file called `server.csr` that looks something like this:

```
-----BEGIN CERTIFICATE REQUEST-----  
MIICLTCCAZYCAQAwgaQxCzAJBgNVBAYTA1VTMRMwEQYDVQQLIEwpcZm9ybmlh  
MREwDwYDVQQHEwhTYW4gSm9zZTEhMB8GA1UEChMYSW50ZXJuZXQgV2lkZ2l0cyBQ  
dHkgTHRkMRYwFAYDVQQLEw1jb21wYW55IHN0b3JlMREwDwYDVQQDEwhKb2huIERv  
ZTEfMB0GCSqGSIb3DQEJARYQdGVzdGluZ0B0ZXN0LmNvbTCBnzANBgkqhkiG9w0B  
AQEFAAOBjQAwgYkCgYEA4eaMiuQxW2voiz0QGdvVMiy8i+4nKCa105z4s5JrH/QP  
JMhE4jqPIIeozCEgDqV+0+0cP9wddVQzDdiOWy3T2337Z/bVCTzKdu0hSHA5Chlk  
B4Xa4AnRF1yOJp6/cipvyXk6gTOAemR91rTRnJ4G8PvDZXq2uz7zhgxtnin+UQkC  
AwEAAaBIMB0GCSqGSIb3DQEJBzEQEw50aGlzIGlzlGEgdGVzdDAnBgkqhkiG9w0B  
CQIxGhMYSW50ZXJuZXQgV2lkZ2l0cyBQdHkgTHRkMA0GCSqGSIb3DQEBBAUAA4GB  
AGsNoOkHw26HYB0Irdx10Q7/3GQ/NX1Sy4BxvaB33zdOSQ5sCnlAx8f6/cKuZe8  
zsFtxrm9zdovolct6zCg+OJxkMGiA+HCyfkrcy+zYBSuqExrzT6Z0Bkub2oRv/ef  
WAi/4kyMf/KfAvCE114ranBbGj9Uk/t7maBLW1+xVxjS  
-----END CERTIFICATE REQUEST-----
```

Again, Windows users should open a DOS console and enter `C:\Program Files\nusphere\apache\` directory. The command for CSR creation is the same on Windows as it is on Linux.

Now for the easy part: go to your Certification Authority of choice, complete their form and pay the fee (usually between \$125 and \$300). Depending on the type of organization involved, you may need to provide documentation via fax regarding company ownership or proof of domain ownership. Usually, the process takes anywhere from twelve hours to a few days to complete, before you can get your SSL certificate.

Installing Your SSL Certificate

Once you receive your SSL certificate, you'll need to place it and your `server.key` file in a place where Apache can find it. These directories are indicated in your Apache configuration file (`httpd.conf`), in the `conf/` directory within your Apache installation directory.

You can name your `server.key` and SSL certificate file anything you want, but these files are usually called `server.key` and `server.crt`. In fact, if you keep these files named as such, you won't have to make any changes to `httpd.conf`. NuSphere automatically places the following directives in your `httpd.conf` file for you:

```
SSLCertificateFile conf/ssl.crt/server.crt
```

```
SSLCertificateKeyFile conf/ssl.key/server.key
```

If you do change the names, please be sure to modify these lines. Otherwise, place your files in the `conf/ssl.crt/` and `conf/ssl.key/` subdirectories in the Apache installation directory, and you'll be ready to go.

Restarting Apache, the Secure Way

Starting Apache with your own security certificates in use is no different than starting it with the dummy certificates, like you did earlier in this article.

On Linux, enter:

```
/usr/local/nusphere/apache/bin/apachectl startssl
```

On Windows, the Start Apache item from the Start Menu enables Apache with SSL support.

At this point you may find that you are prompted for your certificate passphrase before Apache will start. If you're at a remote location and your server needs to be rebooted, you would have a problem on your hands. However, you can use OpenSSL to remove the encryption (and thus the need for the password) on your key.

First, copy your `server.key` file to preserve its original state:

```
cp server.key server.key.org
```

Then, use the OpenSSL binary to remove the encryption:

```
openssl rsa -in server.key.org -out server.key
```

The commands work on both Windows and Linux. Be sure to backup your key files, and to put them back where Apache expects them to be. Once the encryption is removed from the key, Apache will start in secure mode without prompting you for the password.

Preparing for Credit Card Processing

Now that you have a secure server, the next step is to obtain a merchant services account with your bank, and integrate the credit card processing software that you are required to use. There are two major players: CyberCash and Verisign. Verisign acquired the PayfloPro software from Signio some time ago, and Verisign also now owns CyberCash. CyberCash still operates as a separate entity; however, if you are not an existing CyberCash customer, you will be directed to Verisign to register for the PayfloPro product.

Obtaining a Merchant Account

Quite often, your bank will tell you which processing software to use, depending on the alignments they have with one company or another. Both the CyberCash Cash Register and Verisign PayfloPro software packages are equally popular, and both work quite well. However, given the current situation at Verisign, chances are you will be directed to use the PayfloPro product.

Both CyberCash and PayfloPro have development kits for Linux and Windows. PHP has functions specific to CyberCash and PayfloPro available to you, if you configure your PHP installation correctly. There is, however, one exception: there is no stable PHP PayfloPro extension currently available on Windows.

Installing Your Processing Tools

Once you decide which tools to use, you must download and install the software. This software—no matter if you use CyberCash or PayfloPro—is independent of your NuSphere installa-

tion. After completing the registration process and paying the fees to the company of your choice, you will receive download instructions as well as user manuals for the software. Both CyberCash and Verisign have excellent products, manuals, and technical support, which will be available to you along the way.

Once you have the software installed, the next step is to modify your NuSphere configuration in order to activate the necessary functions for either CyberCash or PayfloPro.

Modifying Your NuSphere Configuration

Before modifying your NuSphere configuration, you should check to see what you already have installed by default. The easiest way to do this is through the `phpinfo()` function. There is a file called `testenv.php` in the root directory of your installation that contains the `phpinfo()` function.

Open that file in your browser and you will see a nicely-formatted page of PHP credits and information. On Linux, you will see, at the top, a line of configuration directives, beginning with: `../configure`. Highlight that string, and copy it, because you will need to use it in a moment.

On Windows, the CyberCash extension is enabled by default, and you don't have to do anything extra to your NuSphere installation in order to use the CyberCash functions.

Recompiling PHP with Function Support

If you are a Linux user, you must recompile PHP. That configuration line you just copied a moment ago? You'll add one line to it, to prepare for re-compiling PHP.

If you are using CyberCash, add the following to the configuration line:

```
--with-cybercash[=DIR]
```

Where `[DIR]` is the location of the CyberCash software that you should have downloaded and installed from CyberCash.

If you are using PayfloPro, add the following to the configuration line:

```
--with-pfpro[=DIR]
```

Where [DIR] is the location of the PayfloPro software that you should have downloaded and installed from Verisign.

Once you have added the proper information to the configuration directives list, move to the /usr/local/nusphere/apache/php/ directory and run the configure script:

```
./configure [your options]
```

After the configure script runs its course and returns you to the prompt, type `make`. When this command is finished, type `make install`, to produce the final PHP module, which should now include CyberCash or PayfloPro support.

Once the new module is generated, you must restart Apache in order to load this new module. Once Apache is restarted, re-load the `testenv.php` page in your web browser; you should see the new function support listed. Assuming it is listed, you're now ready to write some scripts.

Using PHP Functions to Process Transactions

Once you have the third-party software installed, your PHP installation modified, and your merchant account all ready for income, you should set up a standard library of functions that you can use throughout your e-commerce scripts. The beauty of open source is that there are many of these libraries and sample scripts available for your use.

Connect and Process with CyberCash

As part of your CyberCash support documentation, you have a developer's API reference. You can use this guide to develop your own libraries for use when connecting to CyberCash, but you can save yourself a lot of time by using Nathan Cassano's CyberClass library (<http://www.zend.com/codex.php?id=115&single=1>). If you download this library, you also have

a test script that you can use. All you need to know are your merchant identification string and merchant encryption key from CyberCash.

A test script using the CyberClass library looks like this:

```
<?

/* Replace "/path/to/merchant_conf" with the location of
your merchant configuration file, created when you installed
the CyberCash Merchant Connection Kit */

test = new cyberclass("/path/to/merchant_conf");

/* Create some variables to hold values used for the transaction.
These values can come from a checkout form of your creation, or
can be hard-coded in your script. */

$order_id = "1111";
$amount = "10.00";
$card_number = "4111111111111111";
$card_address = "111 Main Street";
$card_city = "Anytown";
$card_state = "CA";
$card_zip = "99556";
$card_country = "US";
$card_expiration_date = "03/02";
$cardholder_name = "John Q. Public";

$response = $test->SendCC2_1Server('mauthonly',
    array('Order-ID' => '$order_id',
        'Amount' => 'usd $amount',
        'Card-Number' => '$card_number',
```

```
'Card-Address' => '$card_address',  
'Card-City' => '$card_city',  
'Card-State' => '$card_state',  
'Card-Zip' => '$card_zip',  
'Card-Country' => '$card_country',  
'Card-Exp' => '$card_expiration_date',  
'Card-Name' => '$cardholder_name'  
));  
  
?>
```

When you run this script, you are sending transaction information to the CyberCash transaction server. This server will return one of several possible responses indicated in the CyberCash documentation. The response you're looking for is "success", meaning that the transaction was processed and your account will be credited.

The CyberClass library, combined with a quick glance at the CyberCash Merchant Connection Kit documentation, will provide all you need for successful transaction processing using your NuSphere installation.

Connect and Process with Verisign PayfloPro

The PayfloPro functions in PHP allow you to send information to the Verisign Payment Services servers and to receive a response. These functions only provide the gateway; you should read your PayfloPro software documentation thoroughly for the proper format for the information you are sending, and also the responses you could receive.

The first function you call in your script is `pfpro_init()`, which simply initializes the library (you should have installed this library earlier). The next function is `pfpro_process()`, which sends specific information to the transaction processing server. Finally, use `pfpro_cleanup()` to close the connection.

The tricky function is `pfpro_process()`. This function contains transaction information. A simple script would be something like this:

```
<?
```

```
pfpro_init()
```

```
/* Create some variables to hold values used for the transaction.
These values can come from a checkout form of your creation, or
can be hard-coded in your script. */
```

```
$amount = "10.00";
```

```
$card_number = "4111111111111111";
```

```
$card_expiration_date = "0302";
```

```
$response = $test->SendCC2_1Server('mauthonly',
    array('Order-ID' => '$order_id',
        'Amount' => 'usd $amount',
        'Card-Number' => '$card_number',
        'Card-Address' => '$card_address',
        'Card-City' => '$card_city',
        'Card-State' => '$card_state',
        'Card-Zip' => '$card_zip',
        'Card-Country' => '$card_country',
        'Card-Exp' => '$card_expiration_date',
        'Card-Name' => '$cardholder_name'
    ));
```

```
$transaction = array(USER => 'your_merchant_login',
    PWD => 'your_merchant_password',
    TRXTYPE => 'S',
    TENDER => 'C',
```

```
    AMT    => '$amount',
    ACCT   => '$card_number',
    EXPDATE    => '$card_expiration_date'
);

$response = pfpro_process($transaction, "test-payflow.verisign.com");

if (!$response) {
    echo "Couldn't establish a connection.";
} else {
    echo "Response code was ".$response[RESULT];
}

pfpro_cleanup();

?>
```

When you run this script, you are sending transaction information to the Verisign transaction server. This server will return a response to you, or it will die with the error message "Couldn't establish a connection". The next steps you take in your script should be a direct result of the response you receive, so read your PayfloPro software documentation thoroughly.

Helpful Hints

Many developers make some very basic mistakes when setting up their online stores. Make sure to avoid some common pitfalls by asking these questions:

- Is your checkout form secure? Many developers forget that in order for the information to be encrypted between the user's web browser and your web server, the form itself must

be accessed through the `https` protocol, not just the form action. So, be sure that the URL to the form containing input fields for credit card information begins with `https://`.

- Can your shoppers quickly add or delete items? This is a user interface design issue, but developers should also remember that users will want to delete an item as quickly as they can add one. Adding a simple "delete" link or button to the shopping cart display will go a long way towards keeping your users from developing a sense of frustration.
- Do you provide meaningful error messages? In other words, when the credit card cannot be processed due to insufficient funds, do you echo back the raw error message from the processing server, or do you simply say "We're sorry, but your card cannot be processed at this time. Would you like to try again?" Remember, people are already leery of using credit cards online, don't make the experience any scarier than it needs to be.
- Have you provided adequate customer service information? If you're going to take people's money, you have to make them feel comfortable. Simply reiterating shipping policies, offering return information, and other simple customer service elements will go a long way toward alleviating the fears one might have when sending their credit card information to a source that's totally unknown to them.

About NuSphere Corporation

NuSphere delivers the first Internet Application Platform (IAP) based on open source components, providing an integrated foundation that allows companies to deploy reliable, cost-effective, enterprise-class applications across Windows, UNIX and Linux environments. NuSphere® Advantage is an integrated software suite that pairs the reliability and cost-effectiveness of PHP, Apache, Perl and open source databases with new technology for building business-critical web applications and web services. Based in Bedford, Mass., the company's commercial software services include technical support, consulting and training. For more information, visit www.nusphere.com or call +1-781-280-4600.

NuSphere is a registered trademark in Australia, Norway, Hong Kong, Switzerland, and the European Community; NuSphere and PHPed are trademarks of NuSphere Corporation in the U.S. and other countries. Any other trademarks or service marks contained herein are the property of their respective owners.

MySQL AB distributes the MySQL database pursuant to the applicable GNU General Public License that is available as of the date of this publication at <http://www.fsf.org/licenses/gpl.txt> and all of the terms and disclaimers contained therein. NuSphere Corporation is not affiliated with MySQL AB. The products and services of NuSphere Corporation are not sponsored or endorsed by MySQL AB.