Add a trusted certificate authority to IBM i for PHP 5.6

Issue

PHP 5.6 , new in Zend Server 8, enables peer verification via the OpenSSL default CA bundle. OpenSSL on IBM i, installed as part of the 5733SC1 Licensed Program, does not include a default CA bundle. This can cause a fatal error when doing internet requests from PHP using SSL. For example, a SOAP request to a server using SSL (the address starts with https://) may suddenly begin to fail after upgrading to PHP 5.6.

The message may contain this text: *"certificate verify failed".*

Environment

Zend Server 8 or later, using PHP 5.6 or later, running on any supported version of IBM i. **Please note:** If you are running Zend Server 8.5.x, please replace all instances of ZENDPHP7 or zendphp7 with ZENDSVR6 and zendsvr6.

Resolution

```
Note: It is OK to use copy and paste to copy the PASE commands out of this article into the PASE shell to run
them.  However, the default CA bundle file name can be different depending on the version of IBM i,
so please be careful to fix the file name in the pasted commands before running them.  The file name includes
the path, which is different for each version of IBM i.
```

To begin, please sign on to a 5250 session as QSECOFR or a *SECOFR class user.

Run this command to start the PASE shell:

```
call QP2TERM
```

In the PASE shell, please set the current directory to the PHP bin directory. This will allow you to run PHP commands interactively using the php-cli command:

```
cd /usr/local/zendphp7/bin
```

Please run the following command to test if this problem exists on your server:

```
php-cli -r 'ini_set("display_errors", 1);file_get_contents("https://www.zend.com/");'
```

If the problem exists, this command will show some errors similar to these:

```
Warning: file_get_contents(): SSL operation failed with code 1. OpenSSL Error
messages:
error:14090086:SSL routines:SSL3_GET_SERVER_CERTIFICATE:certificate verify failed in Command line code on line
1


Warning: file_get_contents(): Failed to enable crypto in Command line code on line
1


Warning: file_get_contents(https://www.zend.com/): failed to open stream: operation failed in Command line
code on line 1
```

If you see the errors, please continue with this procedure. The next step is to find the default location for the default CA bundle:

```
php-cli -r 'var_dump(openssl_get_cert_locations());'
```

This will output an array. The first element of the array will show the default CA bundle file name:

```
array(8) {
  ["default_cert_file"]=>
  string(38) "/usr/local/openssl-1.0.1k/ssl/cert.pem"
```

Please use the ls command to verify that this file does not already exist. ***Please be careful to use the actual file name (and path) identified by the previous command***:

```
ls /usr/local/openssl-1.0.1k/ssl/cert.pem
```

The expected message is this:

```
ls: 0653-341 The file /usr/local/openssl-1.0.1k/ssl/cert.pem does not exist.
```

Check to see if the containing folder exists:

```
ls /usr/local/openssl-1.0.1k/ssl
```

If the folder does not exist, you will see this message:

```
ls: 0653-341 The file /usr/local/openssl-1.0.1k/ssl does not exist.
```

If needed, create the folder:

```
mkdir -p /usr/local/openssl-1.0.1k/ssl
```

To create the cert.pem file, you will need to find a valid trusted Certificate Authority bundle file to copy. You will be using this content to identify trusted Certificate Authorities. For the purposes of this article, we are using a file published by the good people at curl.haxx.se, the same people that created the cURL utility. Zend can not guarantee or warrant that this CA bundle is safe for you to use. This is your decision if you want to use this example, or if you want to obtain a trusted CA bundle from some other source. The requirement is that the CA bundle is in the .pem format, and that it is provided by someone you trust. The example fills the requirement for a .pem formatted file, but only you can determine who you trust. So, with that in mind, here is an example of how you might create the cert.pem file using an example source from the internet (remember to use your specific cert.pem file location):

```
php-cli -r 'ini_set("display_errors",
1);
          $context = stream_context_create(["ssl" =>["verify_peer" =>
false]]);
          readfile("https://curl.haxx.se/ca/cacert.pem", false, $context);' > /usr/local/openssl-1.0.1k/ssl
/cert.pem
```

This command can run for a short while, so please remember to wait for the $ or # to appear before proceeding to the next command. In PASE, there is no "input inhibited", so you can run a new command before allowing the previous command to finish.

In the previous command, we set display_errors on to show any errors that might come up if the command is not successful. We also set up a context resource to set the SSL verify_peer option to false. This allows us to access the site without the .pem file, which of course we don't have until we run the command. You can see more SSL Context Options here:

SSL context options

You can use the cat command to view the contents of this file in your PASE shell. It is quite long, so you can also consider using the head command to display just the first lines of it. You can also open the file to view it in an editor, but be very careful not to update it. Here is an example to show just the first 25 lines (remember to use your specific cert.pem file location):

```
head -25 /usr/local/openssl-1.0.1k/ssl/cert.pem
```

This will show you the heading comments, and then the first couple of lines of the first certificate. Notice that the first comment mentions the date the file was created. This file is updated with new certificate data from time to time, so you can redo this process any time you want to obtain a newer CA bundle.

```
##
## Bundle of CA Root Certificates
##
## Certificate data from Mozilla as of: Wed Apr 22 03:12:04 2015
##
## This is a bundle of X.509 certificates of public Certificate Authorities
## (CA). These were automatically extracted from Mozilla's root certificates
## file (certdata.txt).  This file can be found in the mozilla source tree:
```

Finally, please rerun the test to see if Zend can be found via SSL:

```
php-cli -r 'ini_set("display_errors", 1);file_get_contents("https://www.zend.com/");'
```

If everything went well, there will be no errors.

If you haven't already done so, please set up the Random Number Generator: Enable the Pseudo Random Number Generator on IBM i to support SSL

Details

You can learn more about this new feature of PHP 5.6 here:

OpenSSL changes in PHP 5.6.x

As mentioned at the above link, it is possible to disable peer certificate verification, but it is not recommended.

The example CA bundle from cURL is extracted from the CA bundle in Mozilla, which is used in Firefox. The idea is that anyone using Firefox is already trusting this CA bundle, even though they may not be aware of it. You may feel uncomfortable downloading this CA bundle over the internet, which is a reasonable concern. There are available techniques you can use to extract the CA from Firefox yourself, and you can use the internet to learn more about those. The purpose of this article is to provide our customers with a quick resolution to the problem, and the customer may decide to continue to rely on this CA bundle, or they may decide to work on finding another resource to provide the CA bundle.

Another resource we found for a .pem file is Google. They seem to update more often, so if you need to use Google services, you might choose to use their .pem file (remember to use your specific cert.pem file location):

```
php-cli -r 'ini_set("display_errors",
1);
          $context = stream_context_create(["ssl" =>["verify_peer" =>
false]]);
          readfile("https://pki.google.com/roots.pem", false, $context);' > /usr/local/openssl-1.0.1k/ssl
/cert.pem
```

Thanks to Clark Everetts for providing all of the technical information that is used in this article!