Let's Encrypt SSL

Walkthrough for PowerSchool

This is a fairly advanced tutorial and requires direct access to your PowerSchool server, as well as experience using the command line on a Linux or MacOSX computer. This walkthrough is not written for Windows.

Why Let's Encrypt? Check out https://letsencrypt.org/ to find out more about them. It's a free, automated, and open certificate authority. It has sponsorship from major internet companies. Their mission is to "... give people the digital certificates they need in order to enable HTTPS (SSL/TLS) for websites, for free, in the most user-friendly way we can. We do this because we want to create a more secure and privacy-respecting Web."

The advantage of a free SSL certificate for PowerSchool is primarily and obviously the cost (free). Especially for test servers, it can be a benefit to have a valid and publicly trusted SSL certificate for thorough testing, without having to pay extra for each additional test server. There is a disadvantage, because Let's Encrypt only issues SSL certificates for 90 days. They do this to keep the process fresh in the mind of server admins, and also because their vision is for the process to be automated. Until we can figure out how to automate this for PowerSchool, you may need to expect to repeat the process at least every 90 days. Also, the scheduling engine that is downloaded to the scheduler's local machine may not support the Let's Encrypt root certificate.

Until automation is supported for PowerSchool, we need to build it manually. The first time you do this will be tedious, but it will get familiar and more comfortable when you repeat the process a few times. Visit the following website which help you build a Let's Encrypt cert manually: https://gethttpsforfree.com/

We recommend creating a working folder for the steps below. It will hold your account private and public keys, the TLS private key, the CSR request, and your final product, and the new SSL certificate.

Step 1: Account Key

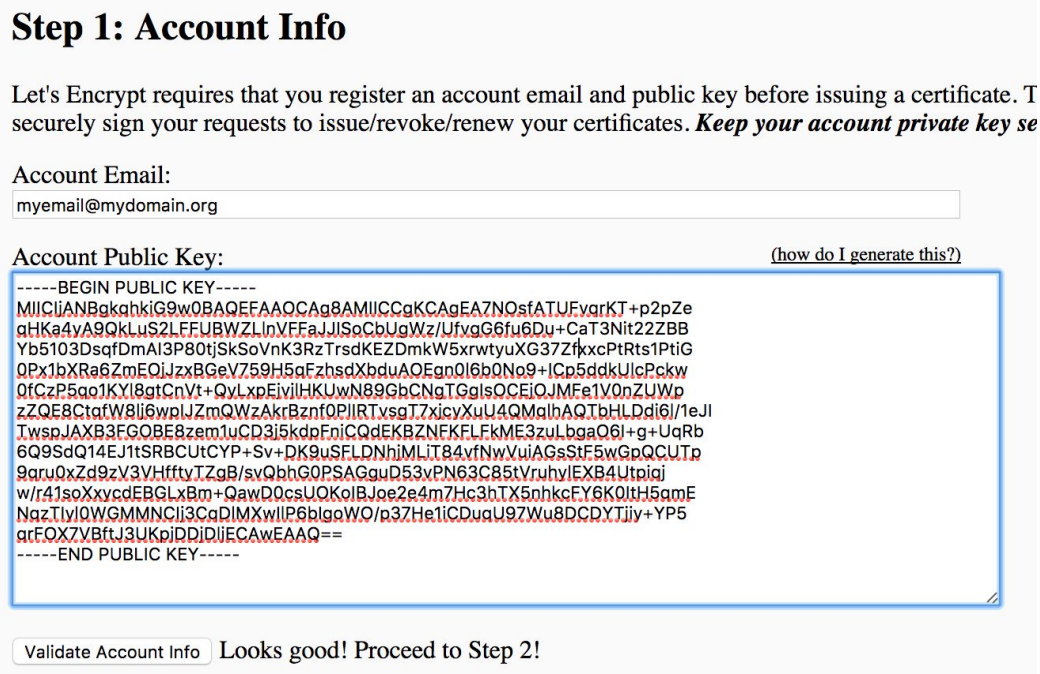
The concept of an "Account Key" is to establish an account for YOU, the certificate requestor. This account key and email is used to issue a cert, renew a cert, or revoke a cert. It is also to contact you if needed. If you have an existing account key, and want to keep a common account across all your Let's Encrypt Certs, then use that common account key whenever you issue/revoke/renew a cert.

Name your private account key "account.key" and save it to your working folder. You'll need it for Step 3.
Name your public key whatever you want, you'll need it to copy/paste the contents in Step 1.

To use an existing account key, enter the email and paste in the PUBLIC key.
(*NEVER, EVER post your PRIVATE key, no one should EVER ask for that. If someone is asking you for your private key, they are an impersonator and are trying to hack your SSL.*)

If you need to generate a new account key, click on "how do I generate this?". These instructions work for a Linux or MacOSX machine via the Terminal. Save the details of your account key in your working folder: the private key (which we recommend naming "account.key"), and the public key.

Click on "Validate Account Info" and get a message - "Looks good! Proceed to Step 2!"



Step 2: Certificate Signing Request (CSR)

A "CSR" is the official request for a certificate. It contains the domain name and the TLS ("Transport Layer Security") public key for your request. Don't confuse this key with the account key in Step 1. This is specific key for this particular CSR. Again, NEVER share the private key, which will be named "domain.key" if you follow the steps.

In our experience, it does not work to use the CSR generator you can find in the PowerSchool Installer, so we generate it manually instead. To generate the CSR, these instructions work for Linux or MacOSX (from the terminal).

For Mac, open Terminal and change to your SSL working folder. If you're not sure how to do this in the terminal, arrange your desktop so you can see both the terminal and finder. Type "cd " (including a space) in the terminal, then drag the folder on top of the terminal command line, and it will complete the request, then just press return.

First generate a private TLS key (not the same as the account private key). Type in Terminal:

```
openssl genrsa 4096 > domain.key
```

Then generate the CSR. The command below is for a Mac, change ps.mydomain.org to the domain name for your PowerSchool server. It's better to format the command in a decent text editor, then copy/paste into the terminal.

```
openssl req -new -sha256 -key domain.key -subj "/" \
  -reqexts SAN -config <(cat /System/Library/OpenSSL/openssl.cnf \
  <(printf "[SAN]\nsubjectAltName=DNS:ps.mydomain.org"))
```

If successful, the terminal will generate a long string which is your CSR. It will look something like the screenshot below:

```
BQADggIBAIS8omFi5mePUGkRRCN9hgKXSmVg05Efdm7TFJ0ZacPI92JGw6X4d4M8
NCwFznYpkdaw3CE9+IhttJHAnzcC6FYYXLBF5O4fHzc2TrZVYY1wW0LhjjrRnAwL
psxqpCs/CALyVib2h1BVwQYGAX0groF8uCow/77fcI6oytSYdAdDeJV8mHjWHsMa
4ZHcKfge/b1QAzs//t4DPaTcyGRu0yhyUL677rwj8NaZyDPNvK8DydGy3chNem+1
zSrJCXlIEvX274nB2jISg0BH1X5d7V7OiVYQWXbcrsSCOo+FLjFnp8Miq1e0HnSZ
uconbD6VaaAFaHjfrySdCugxfnh2S3tXsjVMcexGN8ZO3CrbGXM5BC/5dsDkrUN7
eB3M9gS7sY7kSd3sisaYUoxT4AyLuYAz3BpR9X50uoeIRuU28/Qhu1BeWK6VtL/3
oI/6+wz8PlI+TuvhZcLaGc/J9x6K0/sSMp3rf17iLqMD+UctsBlp2ndAQAjmLcHI
zBmOZw/k+W0dES01EpGWv7WwUNnsXv8mG3CKHnrDB9rGQAden+2/54Pld/6o5UY7
HPZq+4qoX4urOfboTzHOQGukE8flYko3k4e8QdMAlLtxan/pmoYwZvO6VyK8oDHO
1wxZAnJ8kCFM21atYbHUcrnovcUMcIm6xqJ/+gjgDjQlxy1kC1z8
-----END CERTIFICATE REQUEST-----
```

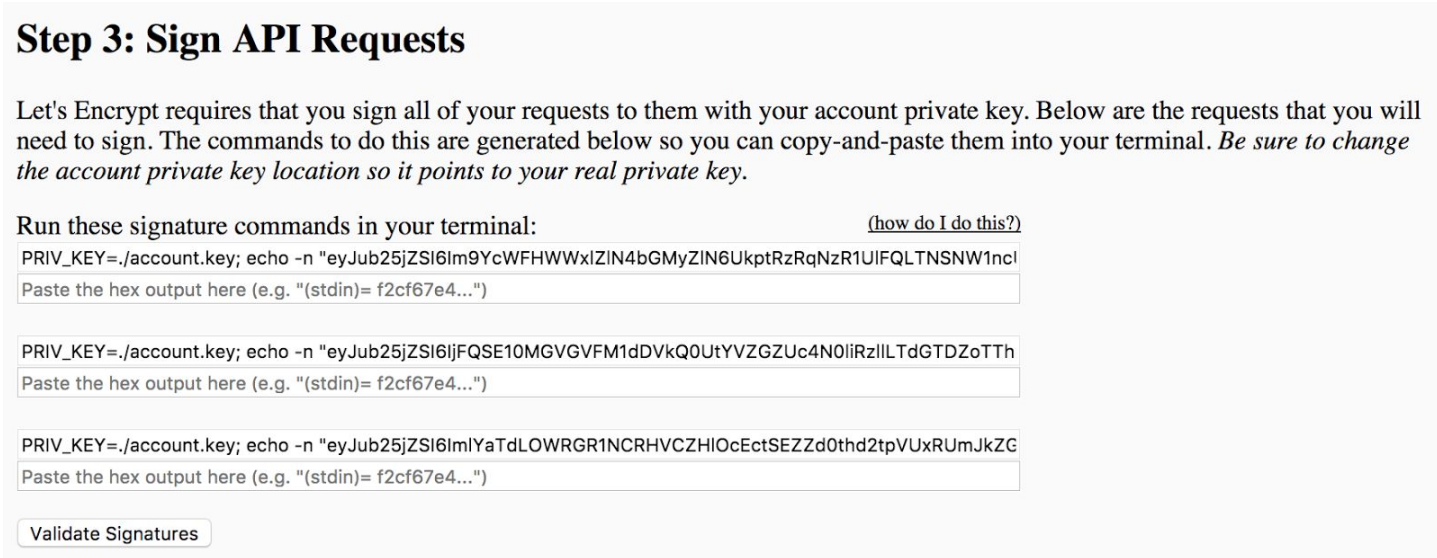
Copy the entire Certificate Request… include the line that says ------ BEGIN CERTIFICATE REQUEST ----- and include the last line that says -----END CERTIFICATE REQUEST -----
You need to paste that text into Step 2. We recommend copying it into a text file and saving it in case you need to repeat any of the process.

Click "Validate CSR", and hopefully you get the message: "Found domains! Proceed to Step 3! (ps.mydomain.org)"

Step 3: Sign API requests

If you've followed the folder and file naming suggestions, you should be able to copy and paste the commands in step 3 into your Terminal and paste the results into step 3 of gethttpsforfree.com

Run the 3 commands (copy/paste into Terminal and press return), copy the results from terminal and paste the output below each command in Step 3, then click "Validate Signatures". If all is good, you'll see the message: **"Step 3 complete! Please proceed to Step 4."**



Step 3: Sign API Requests

Let's Encrypt requires that you sign all of your requests to them with your account private key. Below are the requests that you will need to sign. The commands to do this are generated below so you can copy-and-paste them into your terminal. *Be sure to change the account private key location so it points to your real private key.*

Run these signature commands in your terminal: (how do I do this?)

PRIV_KEY=./account.key; echo -n "eyJub25jZSI6Im9YcWFHWWxlZlN4bGMyZlN6UkptRzRqNzR1UlFQLTNSNW1ncl

Paste the hex output here (e.g. "(stdin)= f2cf67e4...")

PRIV_KEY=./account.key; echo -n "eyJub25jZSI6IjFQSE10MGVGVFM1dDVkQ0UtYVZGZUc4N0liRzllLTdGTDZoTTh

Paste the hex output here (e.g. "(stdin)= f2cf67e4...")

PRIV_KEY=./account.key; echo -n "eyJub25jZSI6ImlYaTdLOWRGR1NCRHVCZHlOcEctSEZZd0thd2tpVUxRUmJkZG

Paste the hex output here (e.g. "(stdin)= f2cf67e4...")

Validate Signatures

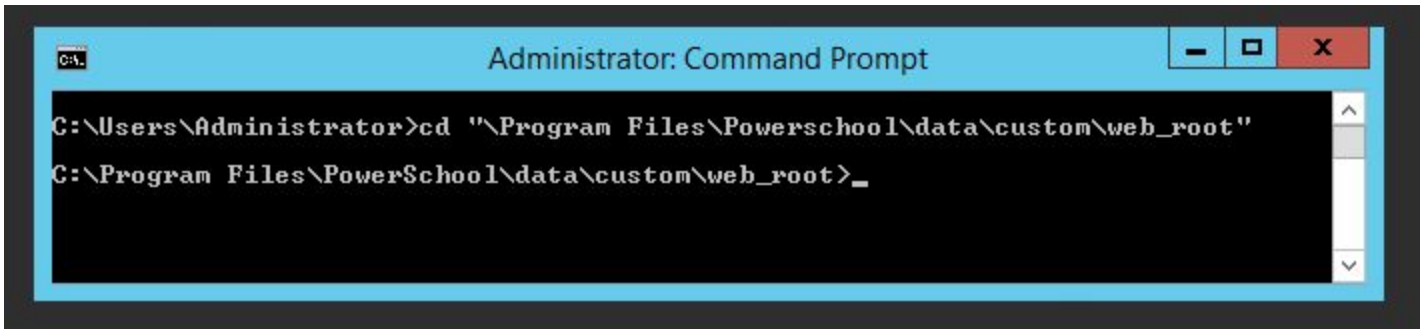
Step 4: Verify Ownership

You need to prove to Let's Encrypt that you actually own this domain. In Step 4 use "Option 2 - file-based" and use direct access to the custom web_root in PowerSchool to host a special file that will prove this to them.

First copy/paste the command and run in Terminal. Paste the results into Step 4 below the command.

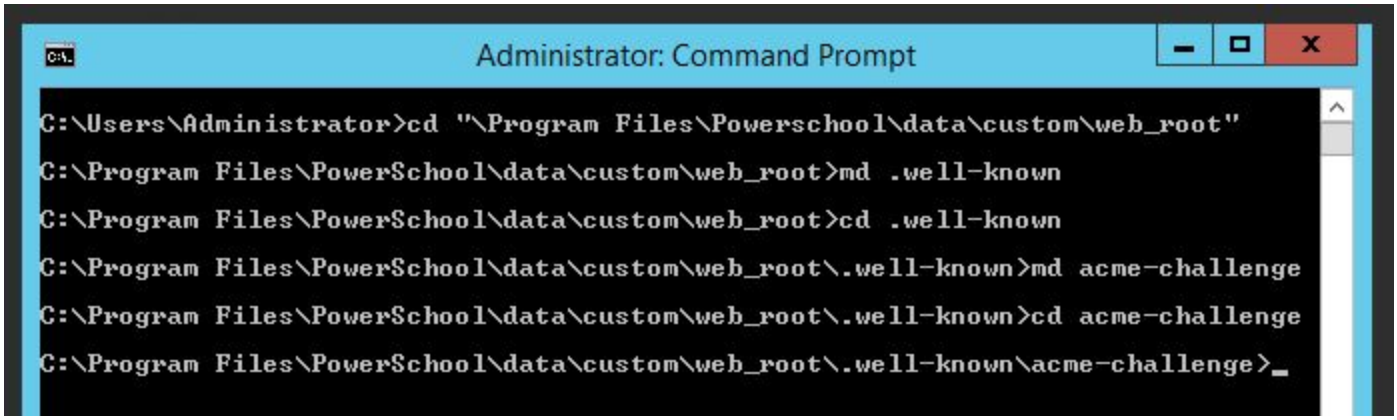
Then for the file-based option, you will need to access your server directly and navigate to the custom web_root from a command prompt. (this will not work in CPM or in Windows Explorer, because they will not accept folders that begin with a ".")

Navigate to the custom web_root folder

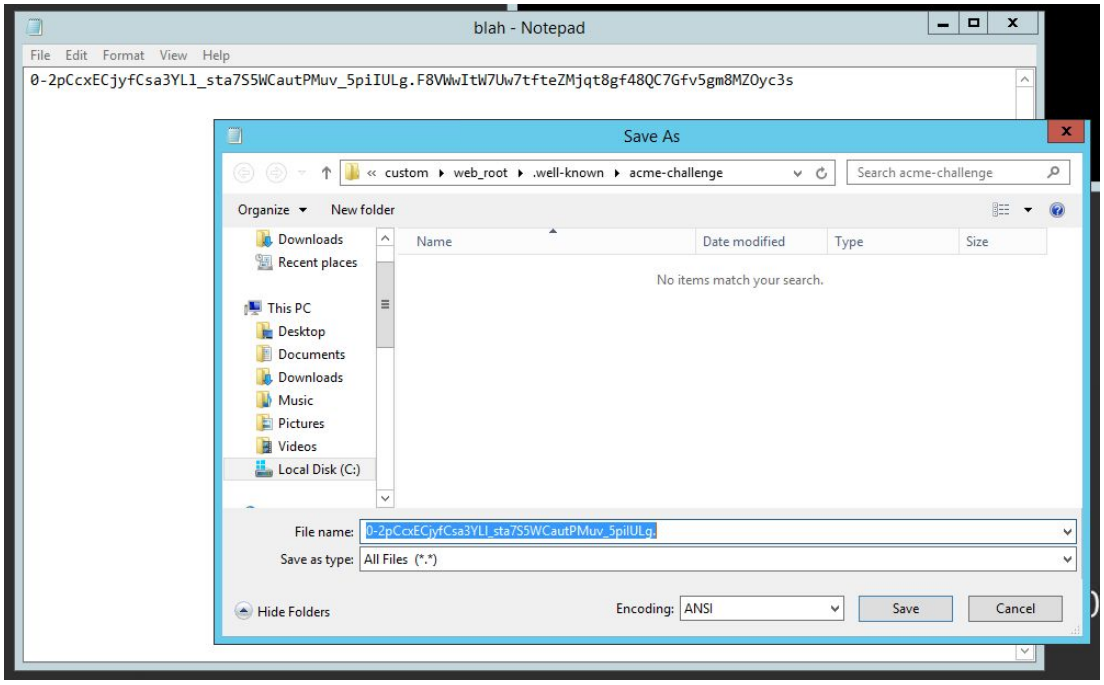


Create a folder at the top level named ".well-known" and change to that folder

Inside that folder create a folder named "acme-challenge" and change to that folder



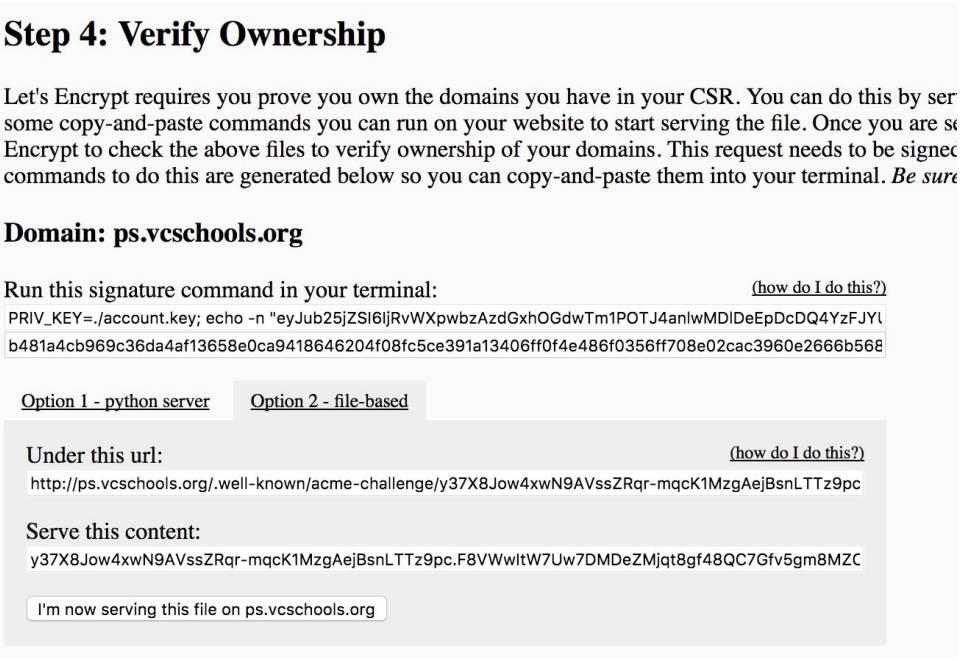
- Open Notepad on the server, you can just type "Notepad" in the command line if you want.
- Copy the "Serve this content" into Notepad, and save it in the acme-challenge folder with the long and scrambled name that is after the http://ps.mydomain.org/.well-known/acme-challenge/
- Note: the filename will have no extension, Notepad will add .txt if you don't specify one, so a trick is to end the filename with just a "." (period), it will then save without a period and without any extension name.



Then either wait 5 minutes or Go to System -> System Settings -> Customization. Uncheck Customization, Submit, Check Customization to re-enable and Submit again. You are temporarily turning off customizations, then turning them back on. This forces the server to update its web_root cache, otherwise you can wait 5 minutes.

Copy and paste the full URL from Step 4 into a browser, make sure it is rendering the scrambled content. Once you're satisfied your server is serving this special file, then in Step 4 click "I'm now serving this file on ps.mydomain.org"

If successful, you'll see the message "Domain Verified!" and you'll see your certs in Step 5.



Step 5: Install Certificate

You should see both a signed Certificate and an Intermediate Certificate in Step 5. You'll need to combine these into one certificate file. use a good text editor like Notepad++ for Windows or BBEdit for Mac and paste both text blocks into one text file. Save it to your SSL working folder. This is the SSL certificate you will install into PowerSchool.

Install in PowerSchool

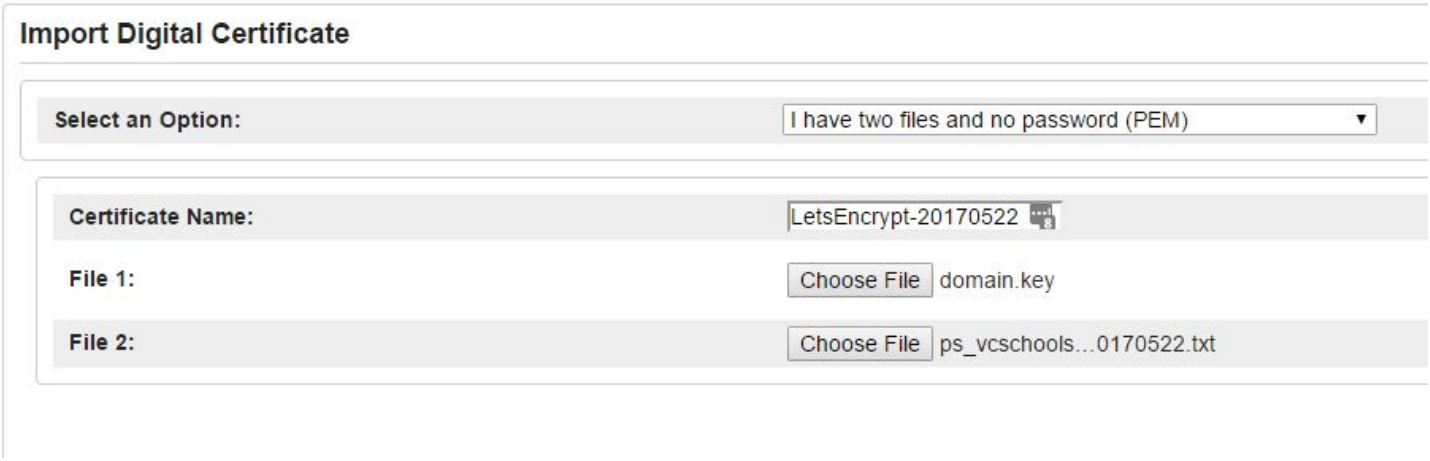
Note: A new SSL cert requires a PowerSchool services restart. Plan for when your users can afford the downtime for the restart and testing.

You need: The combined signed and intermediate certificate from Step 5 and the "domain.key" file from Step 2.

In PowerSchool go to: System -> System Settings -> Digital Certificate Management

- Select an option: I have two files and no password (PEM)
- Certificate Name: (your choice, if you choose the same name as your existing cert, it will not require reconfiguring the PowerSchool installer. However, it may be best practice to save with a unique name - with a datestamp. It will require updating the PowerSchool configuration on the server, but you will still have the old cert in PowerSchool to fall back on if there are issues with the new cert)
- File 1: (attach the domain.key file)
- File 2: (attached the combined signed/intermediate cert you saved in this step)

Click "Import"

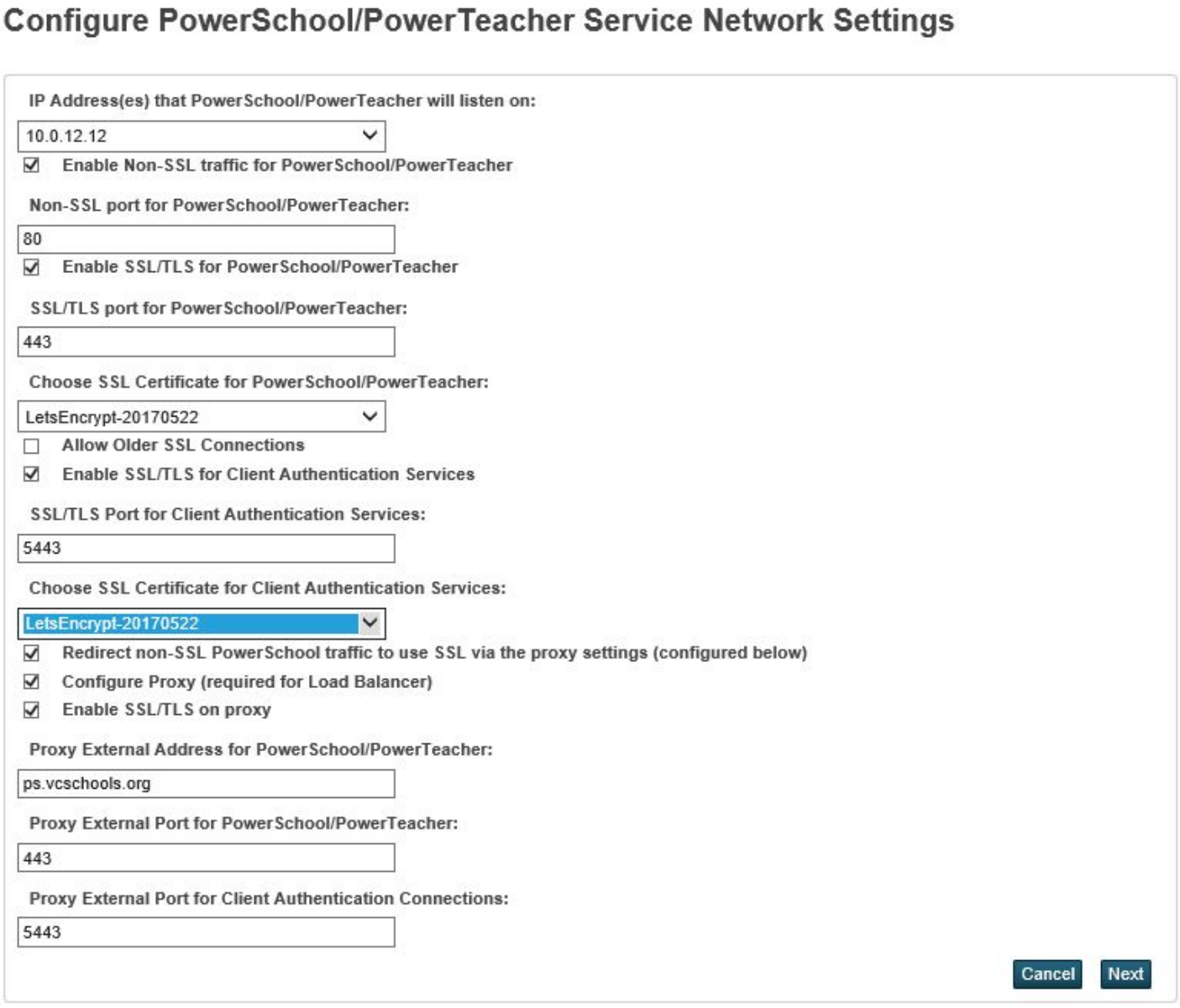


If successful you will see a green "Private Key saved successfully" message at the top and your new cert will show up in the list.

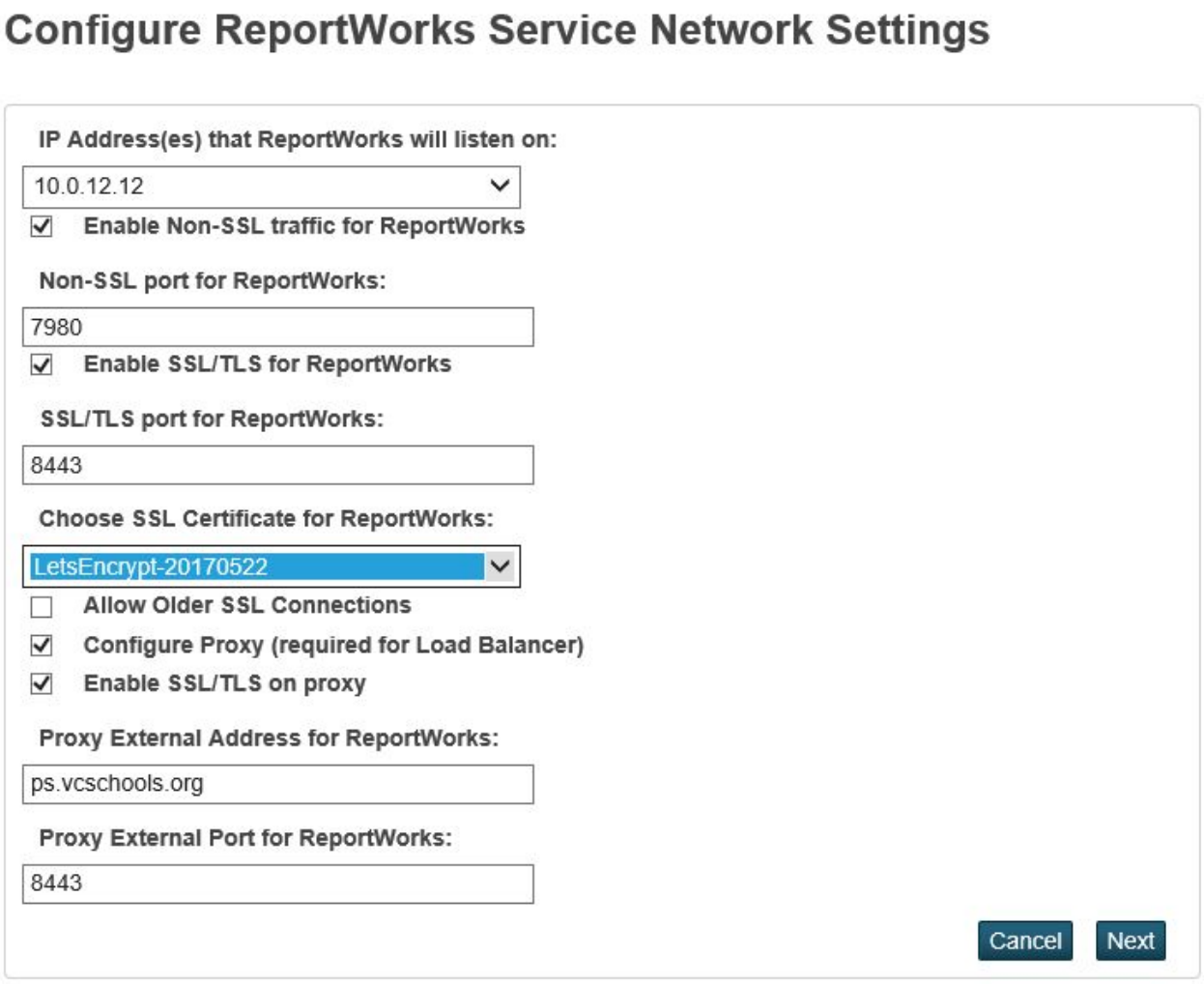
If you chose a new name for the cert, you need to access the server directly and load the Installer page to update the configuration.

You need to install the cert in two areas (*not required if you uploaded the new cert with the same name*):

- **PowerSchool/PowerTeacher:**
    - Choose "Configure PowerSchool/PowerTeacher Service Network Settings". There are two places where you choose the new cert from a list. Click "Next" and follow the prompts.



- **ReportWorks**
    - Choose "Configure ReportWorks Service Network Settings". There is one place to choose the new cert from a list. Click "Next" and follow the prompts.



Final Step - Restarting PowerSchool/PowerTeacher and ReportWorks

If you chose the same name as a previous certificate, you do not need to reconfigure the installer. Either way, you need to restart PowerSchool/PowerTeacher and ReportWorks, which you can do from the Installer Page OR from the PowerSchool admin user interface (System Settings - Reset Server). This will create downtime for your users!!! After the restart, test everything impacted by SSL, including the PTG gradebook and the scheduling engine, if you use it.