CPS 497 Independent Study

App Author: Hannah Seccia

Class Name: CPS 497

Professor: Patrick Seeling

Description: A cross-platform mobile app that connects to a WordPress blog to enable the user to view and create posts.

Date Last Worked On: 3/16/2020

Initial Goal:

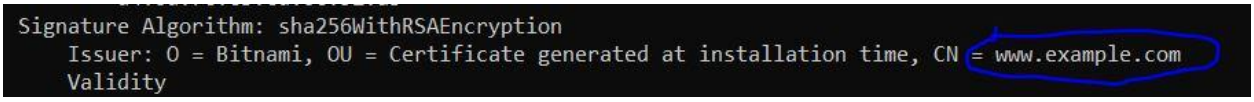
The initial goal of this app was to develop a cross-platform mobile app with Xamarin Forms that serves as an app for a blog. The URL of a WordPress blog was to be connected to with a special WordPress PCL for Xamarin to store information about posts and users in objects. The connection to the blog URL would take place in a splash screen to gather data before the main page was loaded. After the splash screen, the app would load a "newsfeed" of the most recent posts on the site. When a post title is clicked on, the post and details of the post, such as the user that created it, are shown. If the user swiped right on the main page, it would take them to a page with multiple forms to allow the user to create a post. There would be a text field for the post title, post content, and the username of the user. Once the user clicked the button to submit, the post would be uploaded to the blog using the WordPress PCL and connection to the blog.

Process/Setbacks:

The first major issue with making this app was that whenever the PCL tried to request an authentication token from the server, the app would asynchronously hang. Without this authentication, I couldn't do anything involving the blog. I tried to make sure that it wasn't an issue with the username or password sent to the function, but no changes would make a difference.

After weeks of trying to figure out why I couldn't do anything in the app that connected to the server, my error log started to show an error that there was no trust anchor on the certificate chain for the website (for reference, I connected to my WordPress test blog through my local IP/localhost). Upon further research, I realized that the app needed to "pin" a valid certificate to trust the website. Using tools like OpenSSL, I created a self-signed certificate of the Bitnami certificate using its CSR file and key (in PEM form). I then used a custom network security config XML in my Xamarin project and connected the certificate with it. I also installed this certificate on my own Android phone. This made the site accessible from my Android browser when I entered my local IP.

I thought this would get the program to run, but I encountered another error in Visual Studio. The new error told me that the hostname (my local IP in this case) didn't match with anything on the certificate, and therefore wasn't valid. Going on OpenSSL, I could see that, in fact, the common name was set to a dummy example. The certificate wanted me to connect to this domain, even though it was empty.



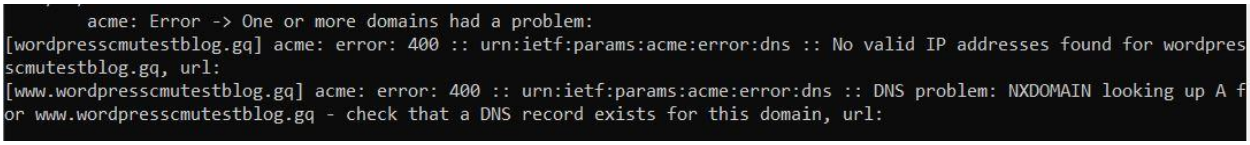
There was no way to modify the common name or alternative names on the certificate. I tried to create a new certificate request with OpenSSL with the common name to my IP, but the WAMP server would not run with this (presumably because it was self-signed, and the information was different). Therefore, the Apache server associated with the stack had to keep its certificate for now.

My next idea was looking up ways to make the app disable its hostname verification since I knew this was possible with Java in Android Studio. Even after hours of research on ways to bypass hostname verification for Android and iOS devices in Xamarin, I eventually realized that since I was using a separate PCL than the standard HttpUrlClient, this was borderline impossible. Also, disabling SSL verification would be incredibly unsafe and enable man-in-the-middle attacks.

I then went back to the idea of a new certificate. Looking into the web for solutions, I found that Bitnami used Let's Encrypt to sign/make new certificates for their stacks, so I tried to make a new certificate signed by Let's Encrypt. I used an application that came with the install (lego), which used their client named ACME. This way, I could set the common name and assign it to my local IP, then hopefully attach this certificate signed by the proper certificate authority to my WAMP. Although, this would prove to be hard with my circumstances.

Let's Encrypt wouldn't make a certificate with IP addresses as the common name since their service didn't allow, as shown by the error log below.

Therefore, I had to find a domain name that would point to the server. I tried to set up a free domain, wordpresscmutestblog.gq, with the site Freenom, a free domain site. I tried both DNS record pointing and forwarding as methods to point to my IP, but in every case, I would get an "invalid IP address" error.



At this point, I had set the project down for the day. When I set the project down and picked it up the next day, I got another variant on the error. The client now timed out when trying to connect.

```
2020/03/15 14:56:46 Could not obtain certificates:
        acme: Error -> One or more domains had a problem:
[wordpresscmutestblog.gq] acme: error: 400 :: urn:ietf:params:acme:error:connection :: Timeout during connect (likely fi
rewall problem), url:
[www.wordpresscmutestblog.gq] acme: error: 400 :: urn:ietf:params:acme:error:connection :: Timeout during connect (likel
y firewall problem), url:
```

When I went to my Freenom domain, my domain had only shown an ad and no longer directed to my local IP. After reading many reviews describing the "shady nature" of free domain sites and Freenom, I decided maybe a domain at a more "secure" hosting site would be ideal. I bought a cheap $0.99 domain off of GoDaddy called "wordpresscmutestblog.com".

After getting this domain, I connected my local IP to this site. During this process, I still got an error, this time a 403 error.

```
2020/03/15 17:05:58 Could not obtain certificates:
        acme: Error -> One or more domains had a problem:
[wordpresscmutestblog.com] acme: error: 400 :: urn:ietf:params:acme:error:dns :: No valid IP addresses found for wordpre
sscmutestblog.com, url:
[www.wordpresscmutestblog.com] acme: error: 400 :: urn:ietf:params:acme:error:dns :: No valid IP addresses found for www
.wordpresscmutestblog.com, url:
```

Turns out that even though I had got a valid domain, it still pointed to my local IP, which was private. Let's Encrypt had to be able to access this website publicly to be able to create a certificate.

My next step after this was to try to port forward my local IP through my router temporarily to make it public. Alas, I still got errors through the ACME client. Even though my IP was technically public, it was showing a 403 error, showing that it still couldn't validate.

```
        acme: Error -> One or more domains had a problem:
[wordpresscmutestblog.com] acme: error: 403 :: urn:ietf:params:acme:error:unauthorized :: Cannot negotiate ALPN protocol
 "acme-tls/1" for tls-alpn-01 challenge, url:
[www.wordpresscmutestblog.com] acme: error: 403 :: urn:ietf:params:acme:error:unauthorized :: Cannot negotiate ALPN prot
ocol "acme-tls/1" for tls-alpn-01 challenge, url:
```

A post on the Let's Encrypt forums said to try using the HTTP challenge instead of TLS on the command line[1]. Instead, the 403 error was simply traded in for another one when I tried the HTTP challenge.

```
        acme: Error -> One or more domains had a problem:
[wordpresscmutestblog.com] acme: error: 403 :: urn:ietf:params:acme:error:unauthorized :: Invalid response from http://w
ordpresscmutestblog.com/.well-known/acme-challenge/C5rv5IVL6TE9Zp0UY0LXbLlUY6yQQOteTDTpcbSwr3M [68.188.174.26]: "<!DOCTY
PE HTML PUBLIC \"-//IETF//DTD HTML 2.0//EN\">\n<html><head>\n<title>404 Not Found</title>\n</head><body>\n<h1>Not Found<
/h1>\n<p", url:
[www.wordpresscmutestblog.com] acme: error: 403 :: urn:ietf:params:acme:error:unauthorized :: Invalid response from http
://www.wordpresscmutestblog.com/.well-known/acme-challenge/FWYOf7obFYaGSKQXjnkNdobczg-cgVo-8-LqR8Tptj0 [68.188.174.26]:
"<!DOCTYPE HTML PUBLIC \"-//IETF//DTD HTML 2.0//EN\">\n<html><head>\n<title>404 Not Found</title>\n</head><body>\n<h1>No
t Found</h1>\n<p", url:
```

Eventually, from many Google searches, I found someone who had the same exact error codes as me on a forum. Their problem was they were using a testing server that points to another server.

1. https://community.letsencrypt.org/t/pn-protocol-acme-tls-1-for-tls-alpn-01-challenge-url-www-peak-codes-acme-error-403-urnparamserror-unauthorized-cannot-negotiat/89980/5

Since the server points to something else by default, it didn't resolve to the right place[2].

At this point, I had hit many walls, errors, and difficulty trying to get a new certificate for my WAMP stack. Most tutorials and pages I had found on the matter had been for cloud-based servers or Lightsail servers, none of which seemed to use your local IP. If there is some magic way to get my local IP onto a certificate, I will be honest and admit that it is much beyond my skill level at this point for web development.

Feeling completely lost, I simply Googled how to connect a localhost WordPress site to a new domain. I have found multiple tutorials that allow this[3]. I have everything for this, except cPanel management. This only comes with buying a hosting plan on GoDaddy, and not just a domain.

Current Progress:

Deciding to not spend money on hosting a GoDaddy server, and not being able to create a valid certificate to be used with the plugin, the app now uses TXT file I/O to simulate the posts and users that would be present on a blog. The app has a prototype splash screen that now creates "test data" and loads it into a TXT file. The splash screen will then sleep for 2 seconds to simulate loading. The main page still loads the most recent post titles by reading from each text file and loading the information into string arrays. When a title is clicked, it will go to a new page and show the post and user who created. When the user presses the back button in the upper left corner, it takes them back to the main page. When they swipe right, there is a prototype post creation page that instead appends the post information to the appropriate text files when the submit button is clicked.

Project Repository:

https://github.com/hseccia/Xamarin_WordPress_IndependentStudy

2. https://community.containo.us/t/not-able-to-obtain-certificates-via-tlschallenge-or-httpchallenge/2288
3. https://www.youtube.com/watch?v=ONFg_2jYCFg