

## Re: Need pointers on managing client certs...

**Source:** <http://www.derkeiler.com/Newsgroups/comp.security.unix/2005-06/0008.html>

---

*kdd21\_at\_hotmail.com*

**Date:** 06/06/05

Date: 6 Jun 2005 11:25:51 -0700

Thanks for the response...

I have been using the "latest" OpenSSL, downloaded about a week ago. And, once I realized I needed to specify the `--cacert` file on the curl command line path, I don't get the message about "certificate verification failed." I was running "curl" without installing it, as I don't want to install it on our build system, so it wasn't finding it in the default path. Problem solved. Or is it?

Based on the text of the error that I was getting:

```
>curl: (60) error setting certificate verify locations:  
> CAfile: /usr/local/share/curl/curl-ca-bundle.crt  
> CApath: none
```

>More details here: <http://curl.haxx.se/docs/sslcerts.html>

```
>curl performs SSL certificate verification by default, using a "bundle"  
> of Certificate Authority (CA) public keys (CA certs). The default  
> bundle is named curl-ca-bundle.crt; you can specify an alternate file  
> using the --cacert option.  
>If this HTTPS server uses a certificate signed by a CA represented in  
> the bundle, the certificate verification probably failed due to a  
> problem with the certificate (it might be expired, or the name might  
> not match the domain name in the URL).  
>If you'd like to turn off curl's verification of the certificate, use  
> the -k (or --insecure) option.
```

This implies that once the server's certificate expires, the local .crt file won't do the trick anymore. Is that true? If so, I presume then the client-side needs to do \*something\* at that point to deal with it, and I'm trying to figure out just what that something is. If not, then that explains why I've been having so much trouble looking for something that I don't need and doesn't exist. If something does have to happen, then it's a darn good thing I inadvertently was running without a default crt file, as I would have otherwise gone blissfully until the massive numbers of "customer down" calls came in because their crt files are all of a sudden out of date...

With browsers, doesn't a user get some kind of popup at that point because his local (whatever they are) are now out-of-date with the secure site he's trying to connect with and the user is at that point asked to make a decision regarding them that he's probably not qualified to make? And that users usually just click on whatever response they think will allow whatever function they are trying to perform work? Frankly, this seems like a fundamental flaw in the design of https security processes— expecting users to understand what is happening in order to make informed decisions about certificates. While giving the decision to the user may let the security design folks off the hook WRT if something happens and a user connects to an invalid site or something and loses his bank account, it seems to me that expecting users to understand what is happening when they get a certificate popup is unrealistic. So far, I'm having trouble figuring out what the terminology means, and I'm \*trying\* to figure out what the \*right thing\* to do about this stuff is, and I've been in the computer biz for 20 years. If I'm having this much trouble, when your Grandmother gets online she's never going to get what's going on with a certificate popup in a browser...

I'd appreciate any further clarifications or pointers anyone can provide...

Thanks,

--  
KD