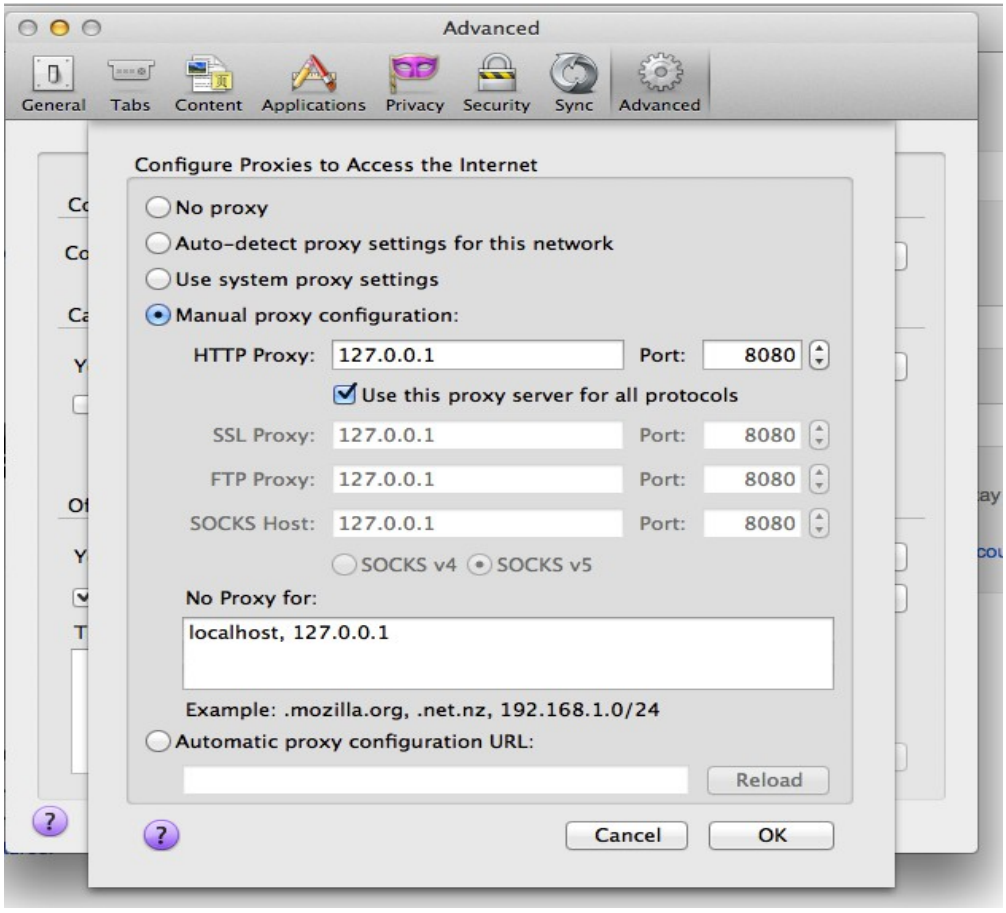
Damn Vulnerable IOS Application Solutions
http://damnvulnerableiosapp.com/

Transport Layer Security – HTTPs

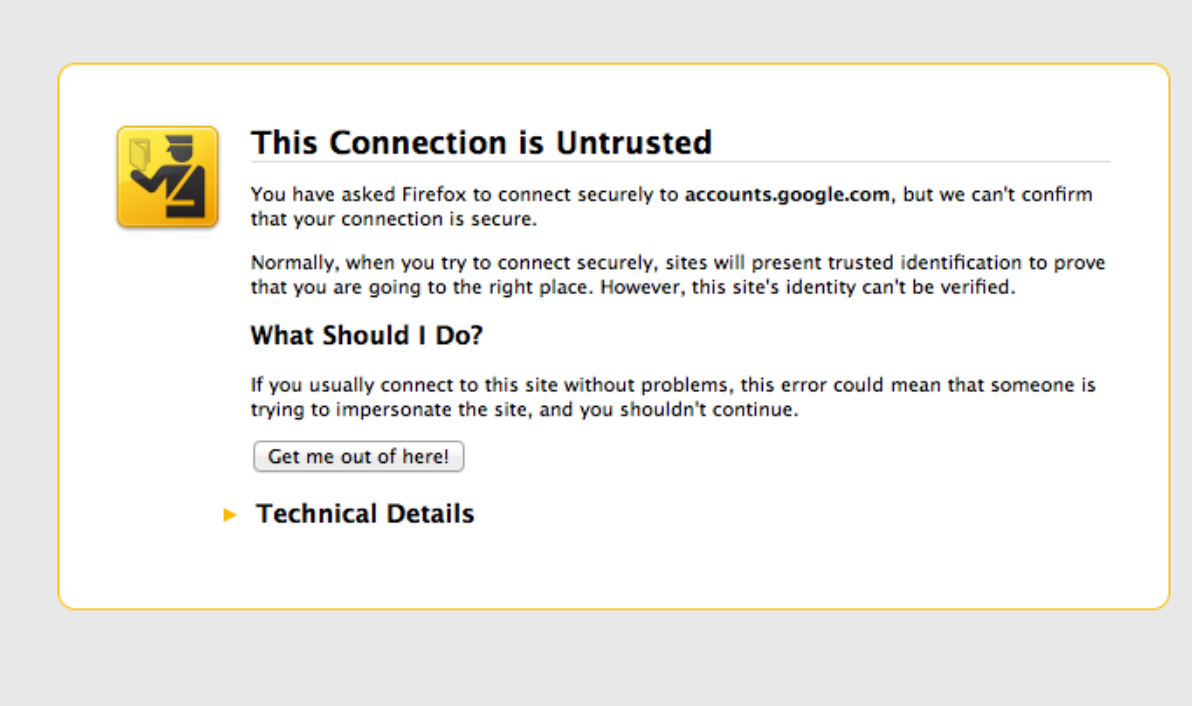
Some applications are coded in such a way that only SSL connections are allowed. However, some applications will issue a warning and prompt to confirm or cancel the connection if the SSL certificate is found be untrusted.

Whenever we connect to a HTTPS website using Burpsuite as a proxy, Burp generates an SSL certificate for each host, which is signed by its own Certificate Authority (CA) certificate. In order to make sure that these warnings don't come up all the time, we have to validate Burp's CA certificate as a trusted root on the device. Hence, the steps would be to first get the root certificate, then install it on the device. Once it is on the device and is a trusted root certificate, it can sign all the certificates and all of them will be treated as valid. Please note that the private key for this certificate is stored in your computer and hence when the traffic passes through the proxy running on your computer, Burp can decrypt the data using its private key. The root CA certificate is created once you install Burp on your system.

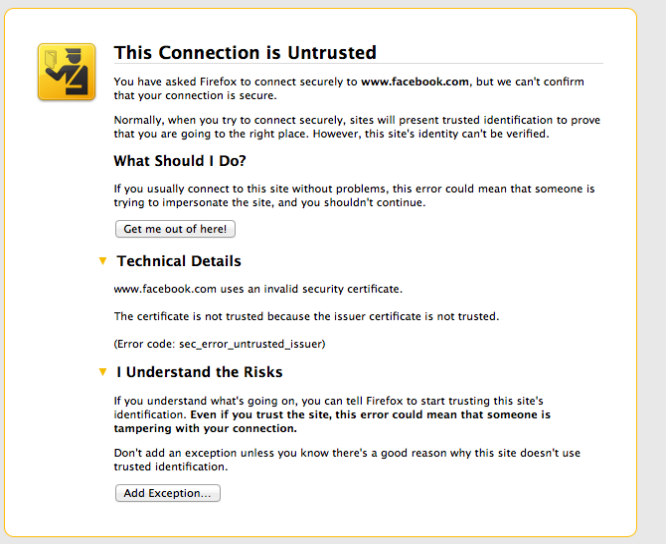
To install the root certificate on your system, first configure your browser to use the Burpsuite proxy.



Then browse to a website that uses SSL (for e.g https://facebook.com). You will be shown a warning

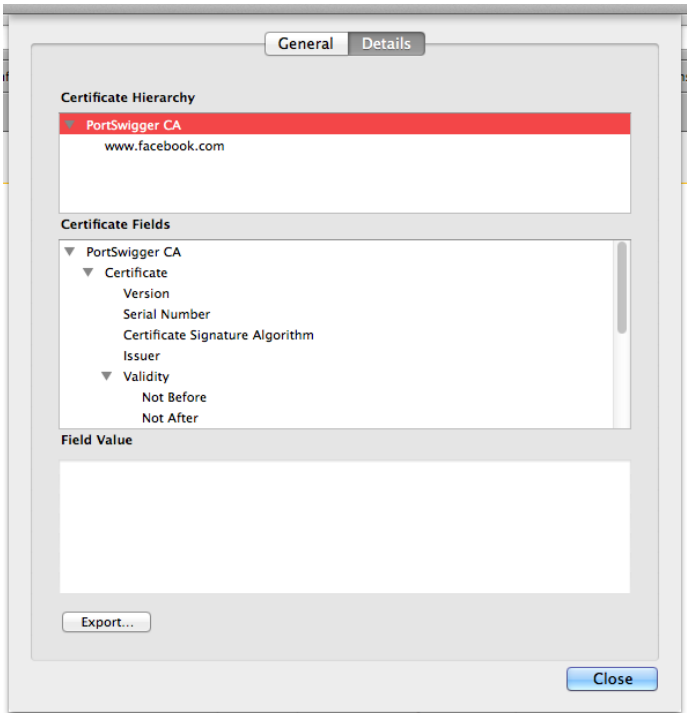


Now our job is to export the root certificate which is used to sign all these certificates. For the domain gmail.com, it is not possible to export the root CA certificate because we cannot add an exception to the gmail domain. Every domain can enforce a policy of this kind. However, facebook allows us to add an exception. Go to facebook.com using Firefox. You will get a warning. Tap on *I Understand the Risks* and tap on *Add an Exception*.



Then, click on *View*

Go to the *Detail*s tab and select the topmost certificate in the hierarchy. This is the root CA certificate. Then click on Export and save the file with an extension of *.crt* .



These steps can also be found on Burp's documentation. Here is a screenshot from http://portswigger.net/burp/help/proxy_options_installingCAcert.html#iphone.

IPhone

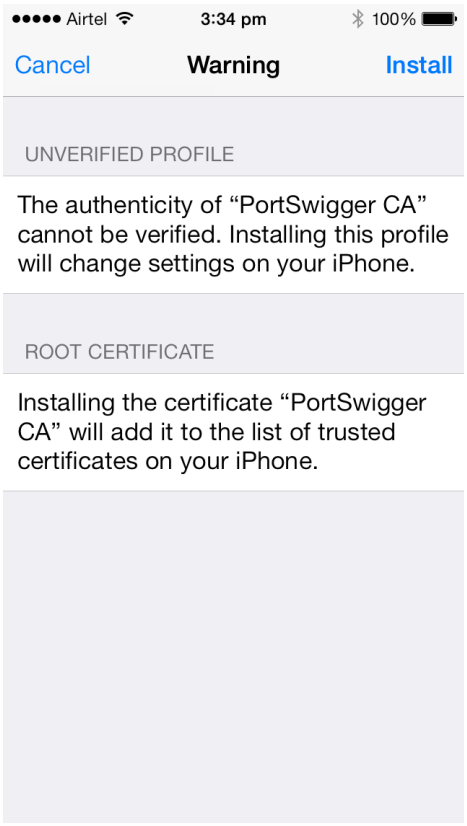
To install Burp's CA certificate on your iPhone or other IOS device, perform the following steps.

1. First, you need to use your desktop browser to export Burp's CA certificate. This is easy with both Internet Explorer and Firefox. Follow the steps to install the CA certificate in your desktop browser, and then visit any HTTPS URL. Click on the padlock / SSL icon to view the details of the SSL certificate. Then select the root certificate in the tree (PortSwigger CA), and in the details for that certificate click the "Export" button. Save the certificate somewhere on your computer using the .crt file extension.
2. Copy the certificate onto your iPhone. The easiest way to do this is to send it as an email attachment from your desktop computer to an account that your iPhone is set up to receive emails for.
3. On your iPhone, open the email and click on the attachment.
4. In the dialog that opens, click the "Install" button, and step through the certificate installation wizard, entering your PIN number if requested.

Note that you may be able to download Burp's CA certificate directly to your device by visiting http://burp/cert with your device configured to use Burp as its proxy.

Now, send this file to your device. Using appropriate social engineering skills, an attacker can get this certificate installed on a device as well without the user knowing its actual consequences. Here is the warning that you get when you open up this certificate. Tap on *Install*

Tap on install again. As you can see, the warning is pretty clear over here.



Tap on done once you are finished.

Now, since this root certificate is treated as valid, every certificate signed by this root certificate will be treated as valid and applications will allow data to be transferred. So now, the traffic going over HTTPS will be intercepted by Burpsuite. As you can see from the figure below, if we enter some dummy credit card details and tap on *Send Over HTTPs*, we can see credit card details being sent through this proxy. You can also see on the top that the url starts with https