# A Sample Man in the Middle (MITM) Attack Against HTTPS

The goal of this exercise is to show you what a MITM attack against you looks like when you are using HTTPS (SSL/TLS). You should find that the certificate error may be the only warning you get that anything is wrong. That’s why it is so important to be aware of the certificate errors the browser gives you.

Unfortunately there are many sites, especially smaller ones, which use expired, self-signed, or misconfigured certificates. In fact, these errors happen much more often than MITM attacks. Certificate errors caused by mistakes may be difficult to distinguish from the errors you get from MITM attacks. A good rule of thumb is, “Never accept a certificate error on a site that has data you care about or that involves money.”

The first stage of a MITM attack is for the attacker to insert themselves in the path between the user (browser) and the web server. On a wireless network, the attacker can set up an access point that has the same SSID the victim uses and overpowers the real access point. On a wired IPv4 network where the attacker controls a computer on the same subnet as the victim, the attacker can use [Ettercap](https://www.ettercap-project.org/) or [Bettercap](https://www.bettercap.org/) to perform ARP cache poisoning to start the MITM attack.

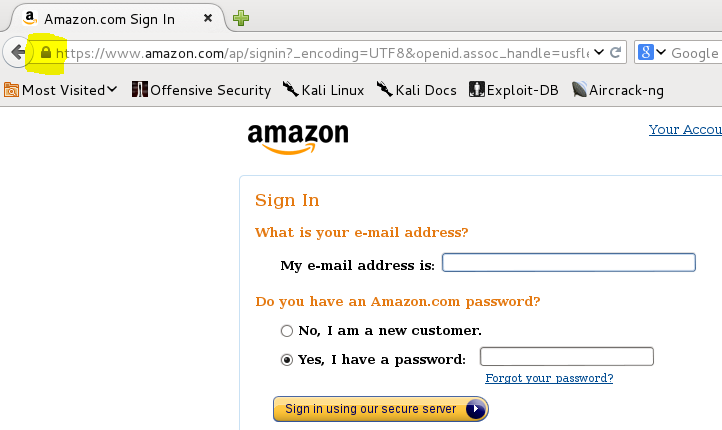


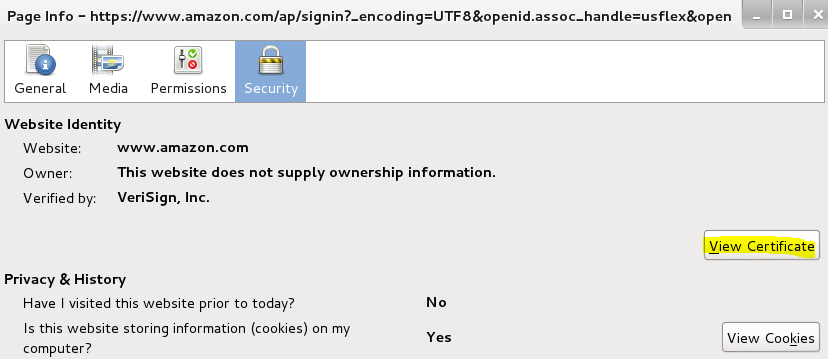
We will assume the first stage has already happened and concentrate on the second stage. In the second stage the attacker intercepts the victim’s traffic with a proxy. Once they’ve done that, they can eavesdrop (steal login passwords, credit card numbers, etc.) or modify the traffic. In an unencrypted HTTP session, this attack can be very difficult to detect. In a session secured by HTTPS, the certificate exchange will fail because the attacker does not have the private key for real web server’s digital certificate. The browser will warn the user, but the user often clicks through the warning (this is bad) and becomes a victim.

Note: If the attacker has stolen the server’s certificate/private key or compromised a legitimate Certificate Authority (CA) that the browser trusts, there may be no warning at all. For more information about an actual attack that involved compromise of a CA and subsequent MITM attacks, Google “Diginotar hack” or “Diginotar Iran.”

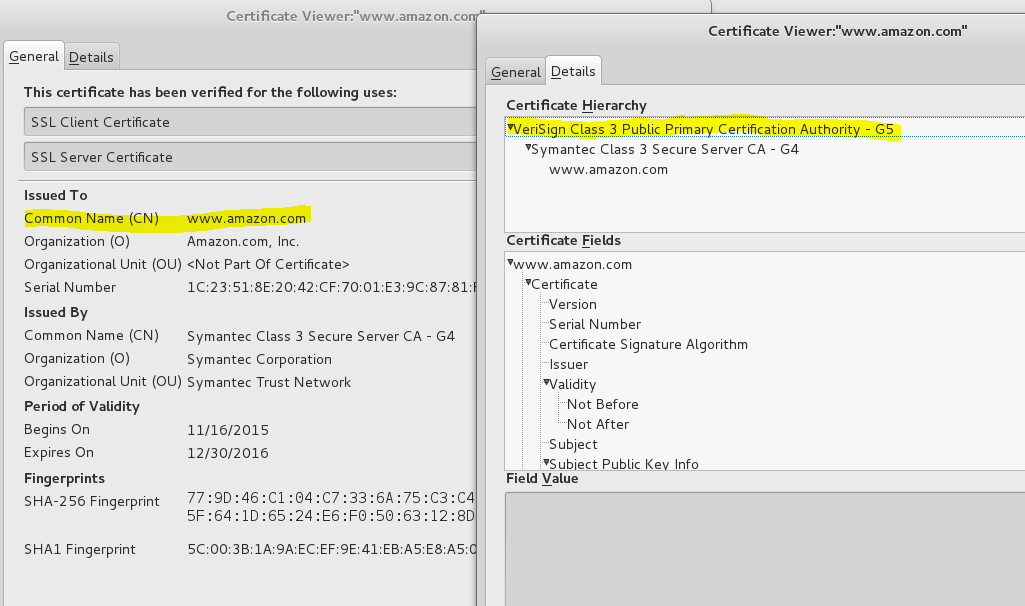
## What’s Normal?

Before we start the attack, let’s see what a valid certificate looks like. Use the Firefox browser in your Kali Virtual Machine (VM) to browse to a site that uses HTTPS and has no errors. I’ll use <https://www.amazon.com> for the “normal” part, but you can use any site that has HTTPS and doesn’t generate certificate errors. Click on the padlock icon to see the certificate information for the site. (Note: Attackers can display a fake padlock icon on older browsers. When you’re making a quick check for encryption, look for https:// in the URL and not the padlock icon.)



Click “More Information” and then “View Certificate.”  
 

Examine both the General and Detail tabs so you know what a valid certificate looks like. Since different browsers show this information differently, you should also examine a valid certificate in your favorite browser.



Notice that the Common Name (CN) on the certificate is [www.amazon.com](http://www.amazon.com). Your browser checks the CN (also the Subject Alternative Names (SAN)) to see that this matches the URL you browsed to, and gives an error if it does not. Also note that the certificate hierarchy starts with VeriSign, which is a Certificate Authority (CA). Your browser has a list of CAs it will accept, along with their public keys. It uses them to ensure that the certificates were actually issued by that CA. Note that in Windows, Google and Internet Explorer use the Windows certificate store which is part of the OS.

Now look at the screenshot of the Windows desktop at the bottom of this article <https://isc.sans.edu/forums/diary/Traffic+pattern+change+noted+in+Fiesta+exploit+kit/19655/>. You’ll see that the attacker is faking the information for a Microsoft certificate. The browser has marked the certificate as invalid because the certificate is not signed by a legitimate CA. Unfortunately, a many users miss this.

## HTTP Strict Transport Security (HSTS)

If the user enters http:// instead of https:// into the browser navigation bar, the web server will often redirect the browser to an HTTPS link. If a MITM attack can intercept this redirection, it can force the browser to stay in HTTP while the attack creates a separate HTTPS connection to the web site. Then the attacker can spy on the connection and the user may not receive error messages. This is the [SSL Strip attack](https://moxie.org/software/sslstrip/) that was published in 2009 by Moxie Marlinspike. Another attack when the browser starts with an HTTPS connection is to give a fake certificate to the browser. In this case, the browser will display certificate errors to the user (but users often ignore the warnings.)

In recent years browsers and web servers have allowed a setting called [HSTS](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security) which greatly increases HTTPS security. If a web site includes the server header “Strict-Transport-Security” the browser will only visit that site using HTTPS and will refuse HTTP connections. The browser caches the HSTS information and future visits will only use HTTPS. This prevents SSL Stripping attacks after the first visit. (There is an option where sites can list themselves on an [HSTS preload](https://hstspreload.org/) site so the first visit is protected as well.)

Additionally, the [HSTS standard](https://tools.ietf.org/html/rfc6797#section-12.1) requires browsers to refuse to allow users to click past certificate warnings if the web site uses the “Strict-Transport-Security” header (no user recourse.) This prevents users from succombing to MITM attacks where the attacker presents a fake certificate. It has the disadvantage that the site becomes unreachable if the web server admin allows the site certificate to expire, however.

HSTS will make our MITM demonstration more difficult if we attempt to visit sites that use the HSTS header. (That’s a good thing.)

## MITM Yourself

Why should you want to hack yourself? Remember that the purpose of this lab is to show what it looks like when an attacker is attempting to MITM your HTTPS session, and to show you that the attacker can see any information in this supposedly encrypted channel. We’re assuming the attacker has already gotten in the middle between your browser and the web site.

We will do the entire lab in the Kali Linux VM. The Burp Suite software will be in the middle, between our browser and the Internet, where the attacker would be. The attacker may very well use Burp as their proxy server, but there are other packages available. Our first step is to tell our browser to send its requests to Burp instead of to the Internet. This simulates that the attacker has inserted themselves into our connection using a bogus access point (wireless) or ARP cache poisoning (wired IPv4.)

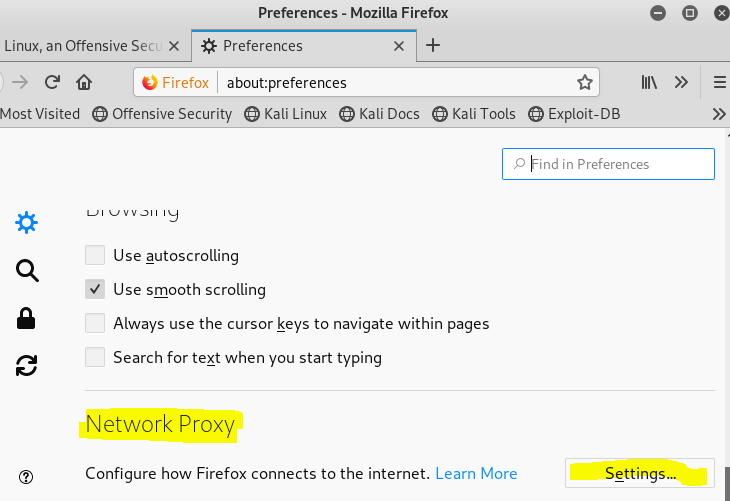
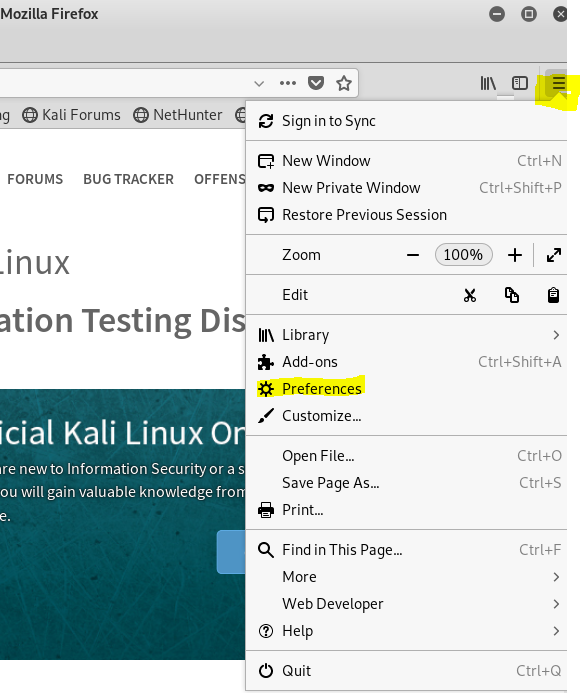
A close up of a logo

Description automatically generated

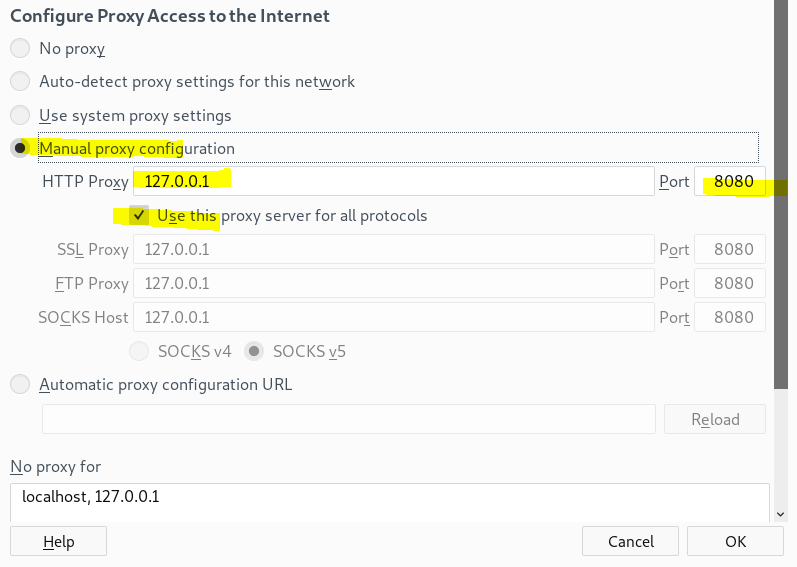
### Browser Settings

**You’ll need to remember to remove these settings when you’re done. Otherwise your browser won’t work after Burp is turned off**.

It’s quite common for testers to pass their web browsers through a proxy on localhost so that they have total control of the information the browser sends to the web site they are testing.

First, open your browser in Kali and use the waffle icon (top right) to select Preferences. Then scroll to the bottom and select Settings for Network Proxy.   


Select Manual proxy and enter 127.0.0.1 (or localhost) and port 8080 into HTTP Proxy, then select “Use this proxy for all protocols.” You can use a different port number if you like, but it has to match the port that Burp is using. Click OK and Close.

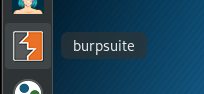
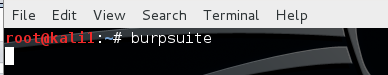


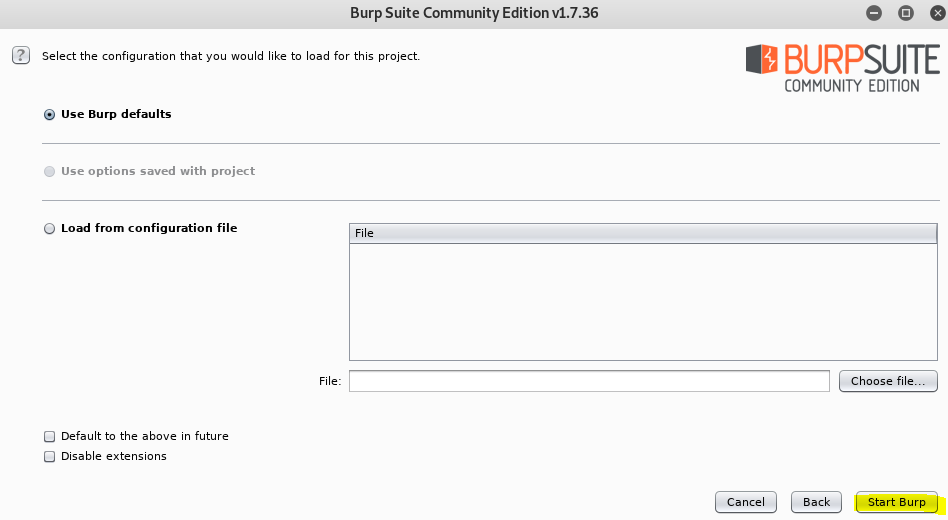
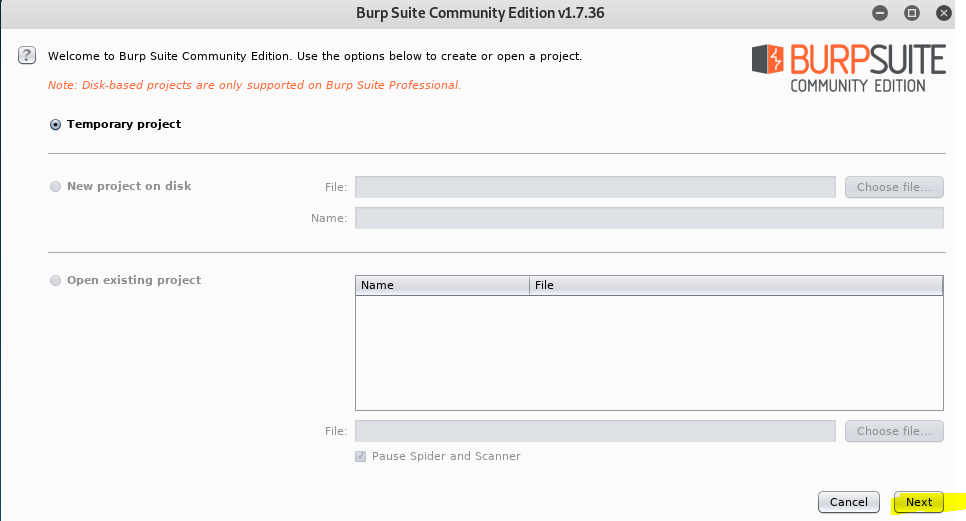
If you try to browse now, you won’t get anywhere because we haven’t started the proxy on Burp yet.

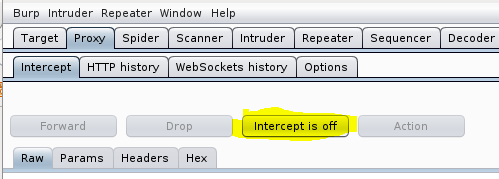
### Burp Suite Settings

The current version of Burp causes SSL\_ERROR\_RX\_RECORD\_TOO\_LONG errors when used with Java v11, which is the default for Kali. To [fix it](https://support.portswigger.net/customer/portal/questions/17434431-gettin-error-code-ssl-error-rx-record-too-long), run   
pdate-alternatives --config java  
and then select Java version 8.

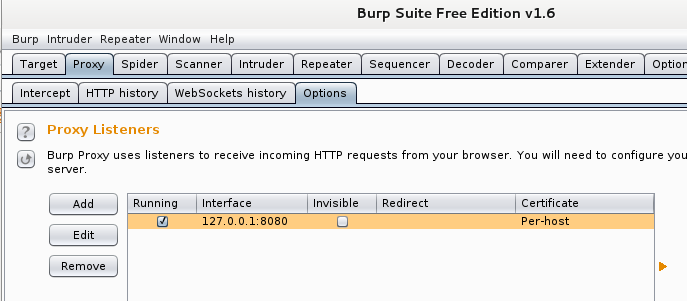


Burp Suite is a full featured web testing tool for penetration testers. We’ll only use its proxy server, which is a small part of its capabilities. You can use the icon, or type burpsuite in a terminal to start Burp.  
 

Click past the first two windows that open.  
  


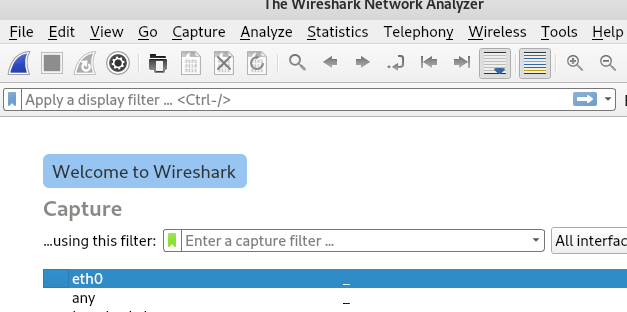
Once Burp opens, select the Proxy tab, and the Intercept tab underneath it. Click the Intercept button until is says “Intercept is off.” When Intercept is on, Burp will stop every packet between the browser and the web site to allow you to make changes. You won’t be altering the packets, so the Intercept option will only slow us down.   
  
 

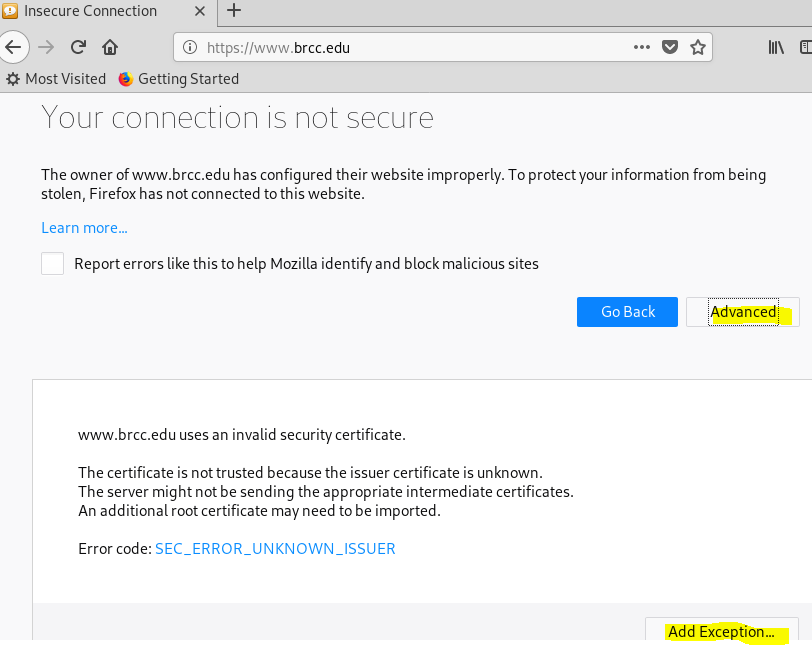
Select the Options tab and make sure that the Burp proxy is listening on localhost port 8080 (127.0.0.1:8080), or whatever port you entered for the Firefox proxy. If you scroll through the rest of this page, you’ll see that there are lots of choices you can make to change the traffic between the browser and the web site.

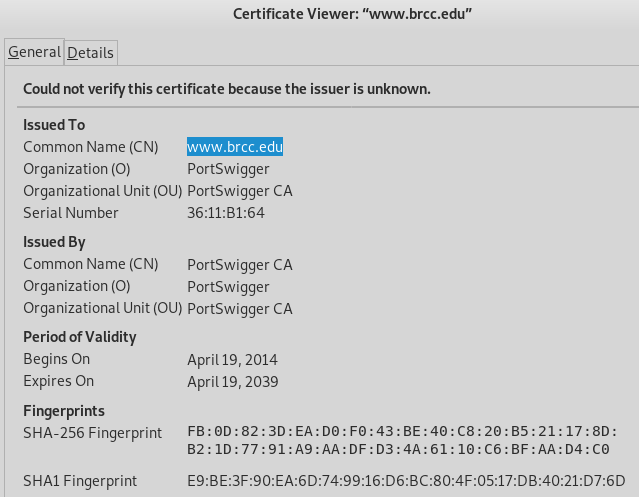
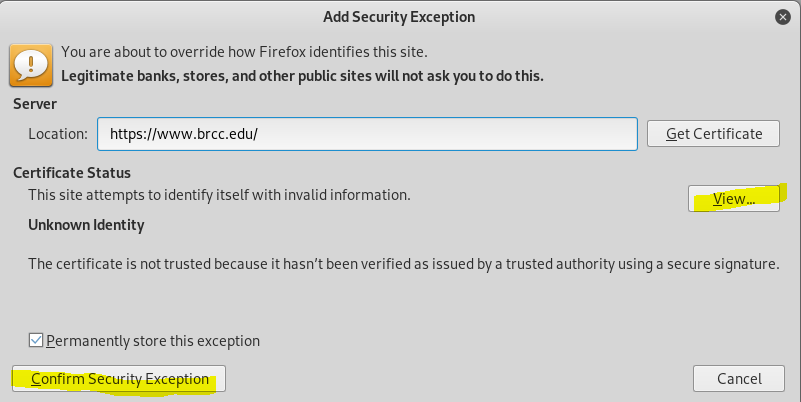


### Attack Yourself

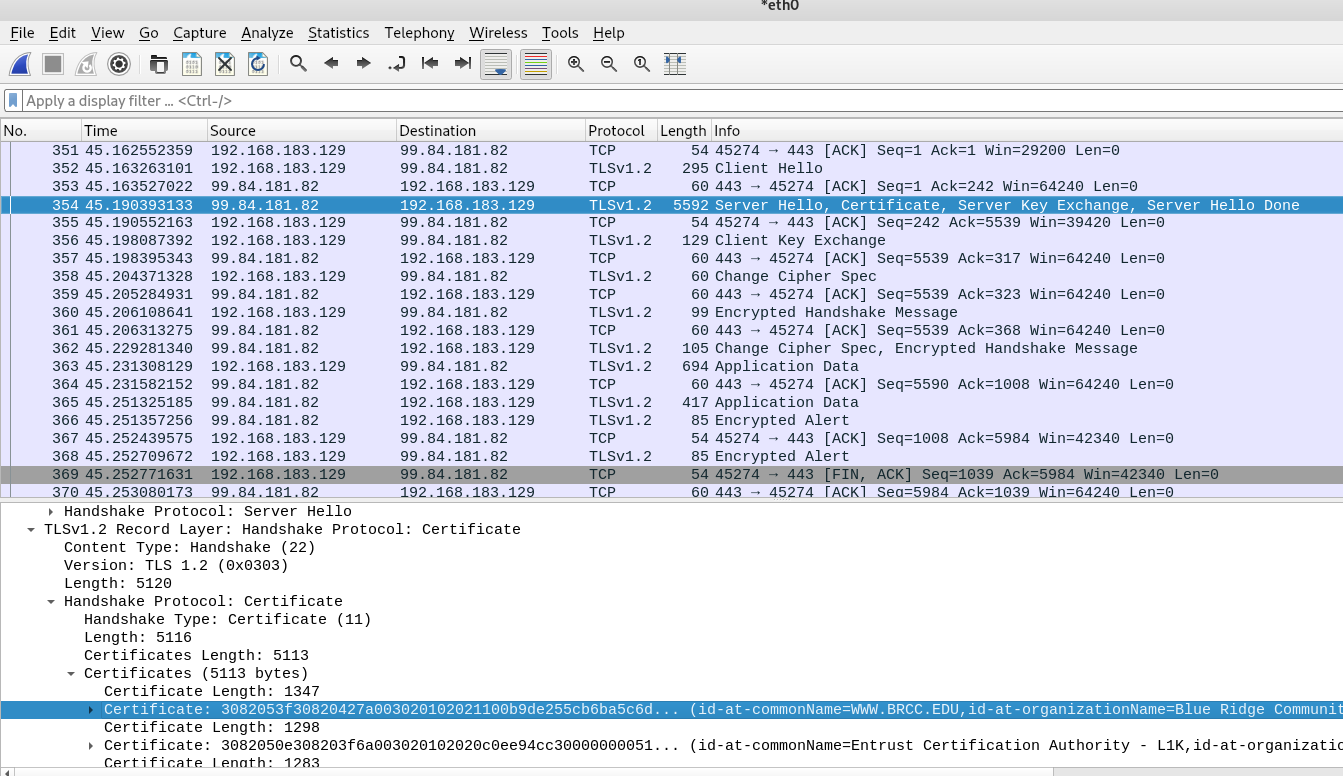
First, start Wireshark, set it to monitor your eth0 interface, and start a packet capture. We will use this to examine the traffic that is going to the web site.



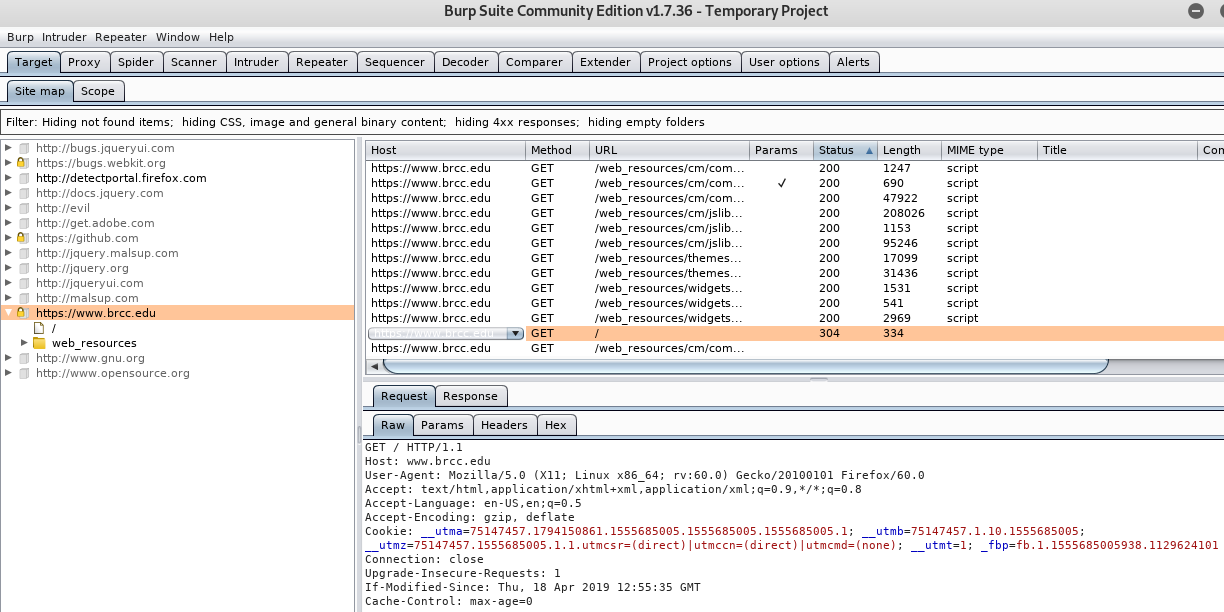
We will use <https://www.brcc.edu/> as a test site, so browse to that from the Firefox browser in Kali. If your browser proxy settings and Burp settings are correct, you should see this. Burp is presenting a fake certificate (we’ll look at it later) and Firefox has detected that the name on the certificate does not match the site we browsed to.  
  


Click the Advanced button (It should really be called “risky” or “dumb”, but whatever.) Firefox explains that the certificate it received for <https://my.vccs.edu/> is invalid, and it is correct. The certificate it received is the fake one generated by Burp, and not the real one. When you view the certificate, you should see something strange; the certificate authority is named PortSwigger. PortSwigger is the company that creates Burp, and it names its bogus CA after itself. We could have created our own CA certificate and named it anything we want, Microsoft for example. The certificate would be invalid, of course, because we don’t have the private key for the Microsoft certificate that our browser trusts. We are brave/foolish, so click to add an exception to allow the invalid certificate.   
  


If you look at the packet capture, you will see that the connection to [www.brcc.edu](http://www.brcc.edu) is encrypted with TLS 1.2. Since we are looking at the eth0 interface, we are monitoring the traffic between the output of the the Burp proxy and the web site. If you use Follow TCP Stream, you will see that the initial connection shows data about the certificates in plain text, but everything else is encrypted.



If you look at the Target tab of Burp, you will see that Burp has decrypted and saved all the content. You can examine the Request and Response of every transaction between the browser and the server.



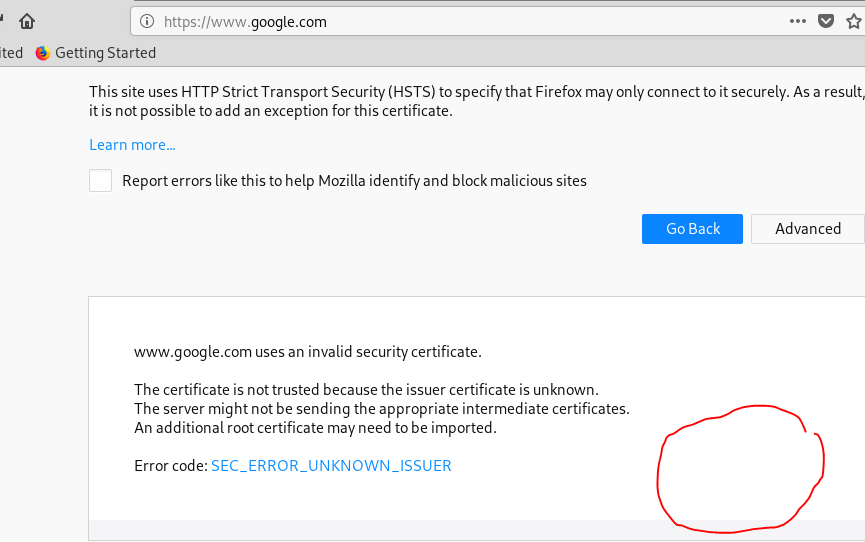
So, our Man in the Middle attacker can see everything that travels between the browser and the server, all because the user clicked past the certificate warning and the web site is not configured to use HSTS. It would be great if we could capture login credentials from the BRCC site (My BRCC in the top left corner), but the vendor the VCCS uses for login is cagey and the login page breaks if you aren’t using valid certificates. You can try it though, for fun.

## Steal Login Credentials

Your instructor will give you a site that does not use HSTS and does not protect its log in credentials as well as the VCCS site. You don’t have valid credetnials so you won’t be able to complete the login. However, you will be able to intercept the user name and password that you enter into the browser!

## Try an HSTS Site

Most major web sites use HSTS now. If you attempt the MITM attack against one of them, you should find that the button that allows you to create an exception (“Add Exception”) is missing.



## Fix your browser

Remember to go back to your Firefox Connection Settings and change back to No Proxy!