HTTPS

- ***1.1: domain name of the website you have set up***
- ***1.1: screenshot of your AWS console where the EC2 instance is visible***
- ***1.1: NGINX configuration file***
- ***1.2: screenshot of Wireshark where it clearly shows a decrypted HTTPS packet***

1.1 Set up a web server and configure HTTPS

The exercise consists of setting up a HTTPs enabled web server from scratch.

- Spin up an Ubuntu t2.micro instance in Amazon Web Services, when you register using your student account, you have gotten free credits (if you have spent all your credits for the assignment, contact me)
- Link a static public IP address to this instance
- Request a free domain name here: https://www.freenom.com/en/index.html?lang=en . This website may be slow and use throttling from the AP source IP address. Use a 5G connection or VPN if you run into problems
- Configure the Freenom DNS so that it resolves to the static public IP address of your EC2 instance
- Connect to this EC2 instance over SSH
- Install NGINX, a famous web server
- Configure NGINX to listen on your domain name
- Install certbot and request a certificate (https://certbot.eff.org/instructions)
- Set up HSTS
- Check the Certificate Transparency logs for your domain: https://ui.ctsearch.entrust.com/ui/ctsearchui
- Check your website against https://www.ssllabs.com/ssltest/ and https://securityheaders.com/ to see if HTTPS was configured correctly

If you are stuck, post a message in the break-out room (during class) or contact me via email (after class).

1.2 Decrypt HTTPS traffic in Wireshark

This exercise may help real-life problems that you face, and as a bonus it should help you understand the inner workings of HTTPS. It involves using Wireshark, a great tool to investigate network traffic. Sometimes using a proxy like Fiddler or Burp to decrypt HTTPS traffic may not be possible or may not be an exact representation of what's happening (for example, Fiddler will automatically downgrade from HTTP2 to HTTP1.1). Wireshark on the other hand gives you a copy of the actual, unaltered network packets and may therefore be more adequate for troubleshooting issues.

However, traffic that is encrypted using TLS will only be decrypted in the application for which the traffic is destined (in the case of HTTPS: your browser). Wireshark will therefore show you the encrypted version. Use the information provided here: https://security.stackexchange.com/questions/35639/decrypting-tls-in-wireshark-when-using-dhe-rsa-ciphersuites/42350#42350 to configure Wireshark in such a way that it is able to decrypt HTTPS traffic.