Password cracking

In the GitHub repo, you will find a file called: “challenge1.zip” – This is an encrypted zip-file, that you need to find the password for.
The password is very weak (between 4 or 6 characters) so it should be fairly easy to brute-force.
Use “fcrackzip” (or any other tool that you like (and makes sense)) to brute-force the password.
When you’ve got the password, you can unzip the content, which is the next challenge (and the information for that challenge).

PGP / GPG email

Encrypt everything!!

A lot of people send messages and e-mails unprotected. This is a very bad idea (in case you were not aware.) Therefor it would be a good idea to start securing the e-mail that you send.

1) Setup PGP, so that you can encrypt your e-mails, by following this guild:
   Windows: https://ssd.eff.org/en/module/how-use-pgp-windows
   Linux: https://ssd.eff.org/en/module/how-use-pgp-linux
   Mac OS: https://ssd.eff.org/en/module/how-use-pgp-mac-os-x
2) Send an encrypted e-mail to someone that uses PGP, and tell them how awesome this is!!

Password manager

We all have so many different accounts for different sites and applications. Sadly, we have a bad habit of choosing a password we can remember and reuse that for all the user accounts we have. This has to stop! The best way to do this is to use a “Password manager”. There are a lot of different once out there. Personally, I’m a big fan of “KeePass” (But this is just my personal choice).

1) Look at different password managers.
2) Start using one from today!!!!

Certificates

Creating a certificate

For this exercise, we will be creating a certificate which we can use for different stuff in our applications. E.g. Encrypt data, identification, etc.
Try to make a self-signed certificate by following the guide in this video:

Video Material: https://www.youtube.com/watch?v=1xtBkukWiek

Encrypted data

In .Net it is fairly simple to encrypt the data send between endpoints – Follow one of these two guide in how to:
Guide 1: https://stephenhaunts.com/2013/03/04/cryptography-in-net-advanced-encryption-standard-aes/

Guide 2: https://dotnetcodr.com/2013/11/14/introduction-to-digital-signatures-in-net-cryptography/

Certificates as permission

In a lot of applications, the user needs to verify their access right with a username and password. But the verification can all be done if we use a certificate to verify that the application/user has access to do different things.
Build a small application (Hello World, a calculator, etc.) that only can be accessed by a user/application if you have the right certificate for it.
There is a lot both text and video guides on how to do this – Find one that works for you.

SSH-keys

Similar to the previous exercise, SSH can also be accessed if we use a certificate. Try to create a SSH certificate and implement it to your GitHub account (After doing this, you don't need to use username and password anymore, when you are communicating with your GitHub account.)
Guide: https://help.github.com/articles/connecting-to-github-with-ssh

TLS certificates (Optional)

When we are browsing the web, it is always a good idea to use a secure connection. For this to happen, the server that we connect to, needs to have a TLS certificate, so that we can establish an encrypted connection between the two endpoints.

Not that long ago, the only way to get a SSL/TLS certificate, was to create one yourself – But, since no one was verifying that homemade certificate, the user would get some "nasty" messages on their screen, which would make them feel not so happy. So, what you would do instead, was to pay a lot of money to have a company to verify that this is a trusted certificate.

Luckily, those days are coming to an end. "Let's encrypt" provides a service where they can generate a trusted TLS certificate for you, for free!! https://letsencrypt.org

1) Install an apache server on your Ubuntu VM.
2) Use "Let's encrypt" to create a TLS certificate and add it to your apache server.