Warm Up

> Submission Deadline: Fri Jan 22 2016 18:00 EST

The goals of this warm-up exercise are to:

1. Establish an account on the class gitlab instance
2. Login to a class container
3. Practice skills you will need to complete subsequent problem sets
    - And, is useful stuff to know anyway!

GitLab

In this class, you're going to be using gitlab to complete your problem sets. So, your first task is to simply activate your account on the class gitlab instance using the login details you should have already received.

SSH Key

We'll be providing containers you can remotely access via SSH to work on your problem sets. In order to access these containers, you will need to associate at least one SSH public key with your gitlab account.

If you don't already have a keypair ready or want one specific to this class, now would be a good time to generate it (preferably using `ssh-keygen`). If in doubt of which algorithm to choose, ED25519 is a decent choice.

A couple of things to note here:

1. **Don’t leak your private key.** It’s supposed to remain secret – that’s why it’s called a private key. So, that means no emailing your private key to me, posting it on piazza, etc. For that matter, it’s best not to generate your keypair on a shared machine.
2. **Use the OpenSSH format.** If you generate a key in PEM, PKCS8, RFC4716, or any other format, you’ll be responsible for converting it.

Once you have your key, associate the *public half* with your gitlab account.

GPG Key

You’ll also need a GPG key. If you don’t already have one, go ahead and generate one. The warning above about keeping your private key private also applies here. But, it is also very important that you **do not lose your private key**. Back it up in a safe place, and perhaps consider using paperkey. Losing your GPG key will cost you points.

Container

Now that you’ve got your keys squared away, you can go ahead and login to your container by ssh-ing to host `svs.wkr.io` on port 800 as user `svs`. If successful, you should be presented with a shell prompt inside your container. You might need to wait a few minutes for your ssh key to become active; be patient!

Repository

Next you'll need to fork the git repository that goes along with this problem set. On gitlab, fork a copy of the repository into your own namespace. You should have a forked repository called `${gitlab_user}/warmup.git`. Once you've done that, clone the fork somewhere on your container.

- Export your GPG key as ASCII and add it to the repository at `gpg/${username}.asc`

Obtain a Secret

A network service is running on `172.17.0.1` port `7789/tcp`. Write a program using the language or tools of your choice that connects to this service to obtain a secret using the following protocol:

```
C->S: L(F(u)) F(u)
S->C: L(v) v
```

where

`C`	Client
`S`	Server
`u`	Your gitlab username in ASCII
`v`	A secret value
`L(.)`	Length of the string argument as a 4-byte integer in big-endian representation
`F(.)`	Byte-wise XOR of the string argument with `0xb7`

- Create a file at `secret/value` in the repository containing *only* the value you obtained from the service

- Add the code you used to obtain the secret to the repository under `secret/`
- Create a script that compiles your program (if necessary) and runs it, assuming the same runtime environment as your container
- Commit the above to your fork

Submission Instructions

- Create a *signed tag* (`git help tag` , see `-s`) indicating the commit representing your submission
- Push your commits and signed tag to your forked repository
- Submit a merge request from your fork to the original repository

Wrapping Up

So, what did we accomplish with all of this? Since this is the general workflow for submitting answers to the problem sets, you've now demonstrated that you're able to work on future assignments. You now have secure access to your container that you can feel free to use for the remainder of the class. And, you've set up cryptographic tools that allow you to strongly assert authorship of your answers, when they were submitted, and that they haven't been tampered with.