Login with password 12345

```
ssh amal@52.55.92.170
```

or

```
ssh aabha@52.55.92.170
```

Change your password

```
passwd
```

TensorFlow Machine Learning

Setup a TensorFlow beginner development environment

```
python3 -m venv tf_env
source tf_env/bin/activate
```

```
pip install -U pip
pip install -U tensorflow-cpu
pip install ipykernel
python -m ipykernel install --user
```

To test if it installed

```
python3 -c "import tensorflow as tf; print(tf.reduce_sum(tf.random.normal([1000, 1000])))"
```

Info log should print two lines and then

```
tf.Tensor(-289.25497, shape=(), dtype=float32)
```

Get the beginner's example demo

```
mkdir tf
wget -P tf https://storage.googleapis.com/tensorflow_docs/docs/site/en/tutorials/quickstart/beginner.ipynb
```

Launch the Jupyter Notebook web server. Choose a port not already being used

```
jupyter notebook --port 1978
```

From the log output, highlight and copy the web server's URL and token

```
http://127.0.0.1:1978/tree?token=10bc6fee24b8c2f9bfe1b135b768923ff05110d1e8ff5136
```

In a web browser, enter the URL but replace `127.0.0.1:1978` with our AWS host `52.55.92.170` and the port `1978` with your port

In the Jupyter web app, navigate to the directory you made, `tf`

Open `beginner.ipynb`

In the menu bar, choose Kernel > Restart & Clear Output

Make sure first "cell" is selected (you'll notice the blue or green highlight on the left)

Step through the Notebook by hitting the Run button. Brackets that don't immediately get a number and instead have a star require waiting time

When it's done, go back to the terminal and Ctrl-C break the server

To leave the development environment

```
deactivate
```

Side-Channel Attacks Assisted with Machine Learning

I already downloaded the repository, datasets and models to your home directories, and made necessary edits

Setup a SCAAML development environment

```
python3 -m venv scaaml_env
source scaaml_env/bin/activate
```

Build it in this order. Their GitHub readme isn't clear with this, but it is a useful reference

```
cd scaaml
pip install -U pip
pip install ipykernel
python -m ipykernel install --user

pip install --require-hashes -r base-tooling-requirements.txt
pip-compile --allow-unsafe requirements.in --generate-hashes --upgrade
python3 -m pip install --require-hashes -r requirements.txt
python setup.py develop
```

That last one exits on an error. Hasn't been a problem

Launch the Jupyter Notebook web server. Choose a port not already being used

```
jupyter notebook --port 1978
```

From the log output, highlight and copy the web server's URL and token

```
http://127.0.0.1:1978/tree?token=10bc6fee24b8c2f9bfe1b135b768923ff05110d1e8ff5136
```

In a web browser, enter the URL but replace `127.0.0.1:1978` with our AWS host `52.55.92.170` and the port `1978` with your port

In the Jupyter web app, navigate to the directory `scaaml/scaaml_intro`

Open `key_recovery_demo.ipynb`

The TinyAES attack will be described

In the menu bar, choose Kernel > Restart & Clear Output

Make sure first "cell" is selected (you'll notice the blue or green highlight on the left)

Step through the Notebook by hitting the Run button. Brackets that don't immediately get a number and instead have a star require waiting time

When it's done, go back to the terminal and Ctrl-C break the server

To leave the development environment

```
deactivate
```

Backup notes - You shouldn't need these, it's just for the setup I did

aws.amazon.com
ubuntu 22.04 jammy amd64 server 2023-09-19 t2.xlarge 4 cpu 16 gb, 300 GiB gp2

/etc/hostname
SCAAML-GMU-ECE-646
adduser
sshd_config

```
apt update
apt upgrade
apt install -y python3-pip python3.10-venv
apt install -y unzip zip
apt install -y jupyter
```

/etc/ssh/sshd_config
Password
systemctl restart sshd

```
jupyter notebook --generate-config
~/.jupyter/jupyter_notebook_config.py
c.ServerApp.allow_origin = '*'
c.ServerApp.ip = '0.0.0.0'
```

I already made the required configuration file in `~/.jupyter/jupyter_notebook_config.py` under your home directory.

https://www.tensorflow.org/install

```
pip install ipywidgets
#pip install -U 'protobuf>=3.4.0'
```

```
git clone https://github.com/google/scaaml.git
```

```
edit requirements.in : tensorflow-cpu

    "import logging\n",
    "import os\n",

    "os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'\n",
    "os.environ['AUTOGRAPH_VERBOSITY'] = '0'\n",
    "logging.getLogger('tensorflow').setLevel(logging.FATAL)\n",

import logging
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'
os.environ['AUTOGRAPH_VERBOSITY'] = '0'
logging.getLogger('tensorflow').setLevel(logging.FATAL)
```

The readme incorrectly says to unzip these files to the `scaaml_demo` directory. I placed them in the `/srv/scaaml` directory from where you may unzip them to the `scaaml_intro` directory that you're in

```
unzip /srv/scaaml/datasets.zip
unzip /srv/scaaml/models.zip
```

```
wget -P /srv/scaaml/ https://storage.googleapis.com/scaaml-public/scaaml_intro/datasets.zip
wget https://storage.googleapis.com/scaaml-public/scaaml_intro/models.zip
```

If you just want to see the output and not run SCAAML, I saved the webpage and placed it in our share directory as `key_recovery_demo.html` and `key_recovery_demo_files/`