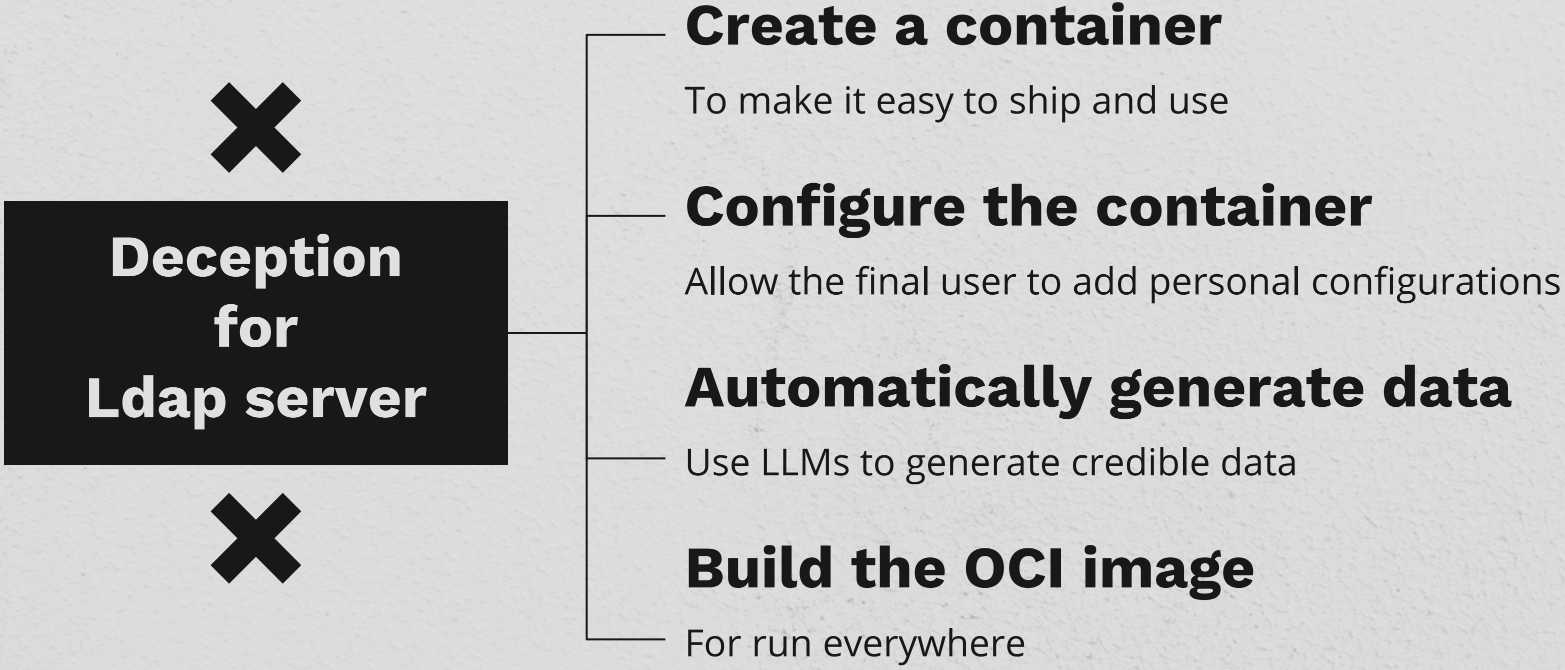
Deception Component Generator

LDAP server

Student:
Gianmiriano Porrazzo

Main goal



Deception for Ldap server

Create a container

To make it easy to ship and use

Configure the container

Allow the final user to add personal configurations

Automatically generate data

Use LLMs to generate credible data

Build the OCI image

For run everywhere

00

DECEPTION FOR DEFENCE

A different way to protect

Deception

Provide false or misleading, but realistic information to the attacker

Denial

Create uncertainty about the real environment that the attacker is facing, to slow down the attacking operations

01

LDAP

Lightweight Directory Access Protocol

Protocol

Defined by IETF

Lightweight

Designed to be a light alternative to DAP

Access

For accessing data stored here

OpenLDAP

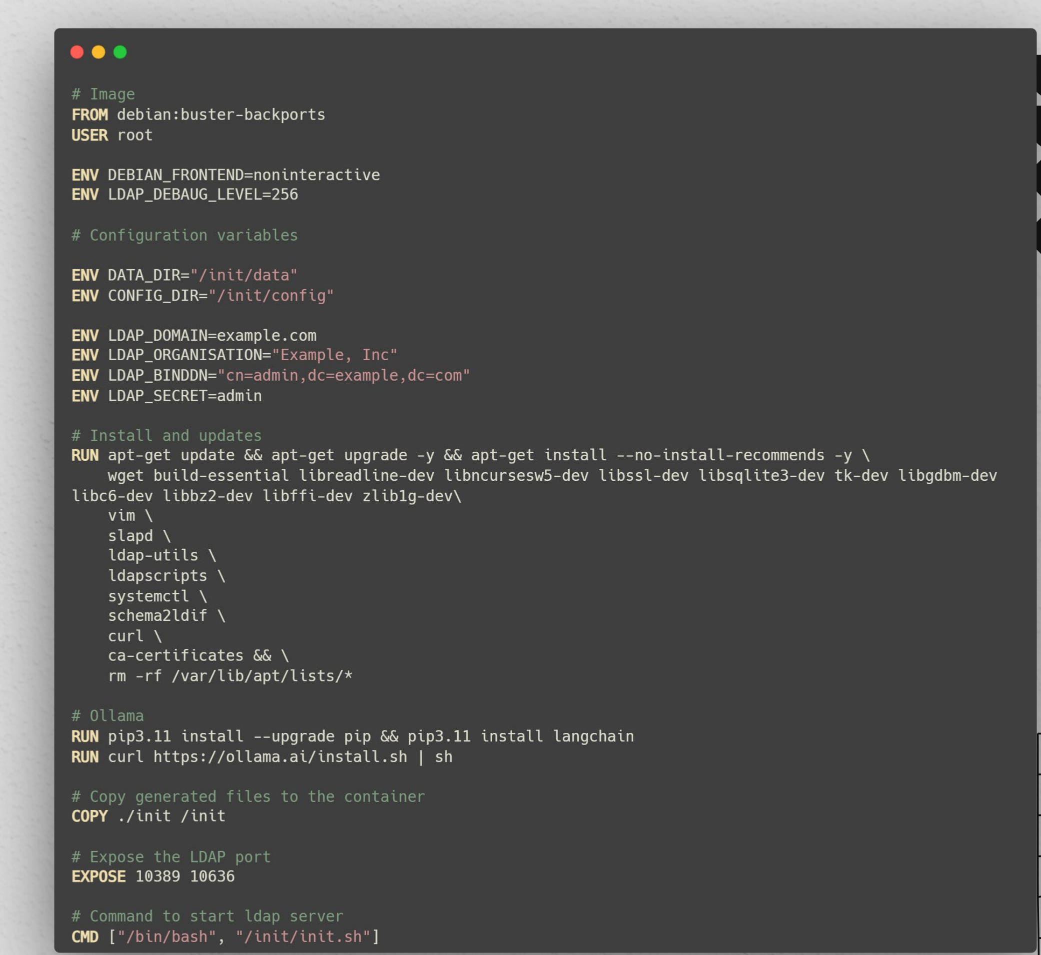
Between various implementation of the protocol this is the one choosen

02

DOCKER

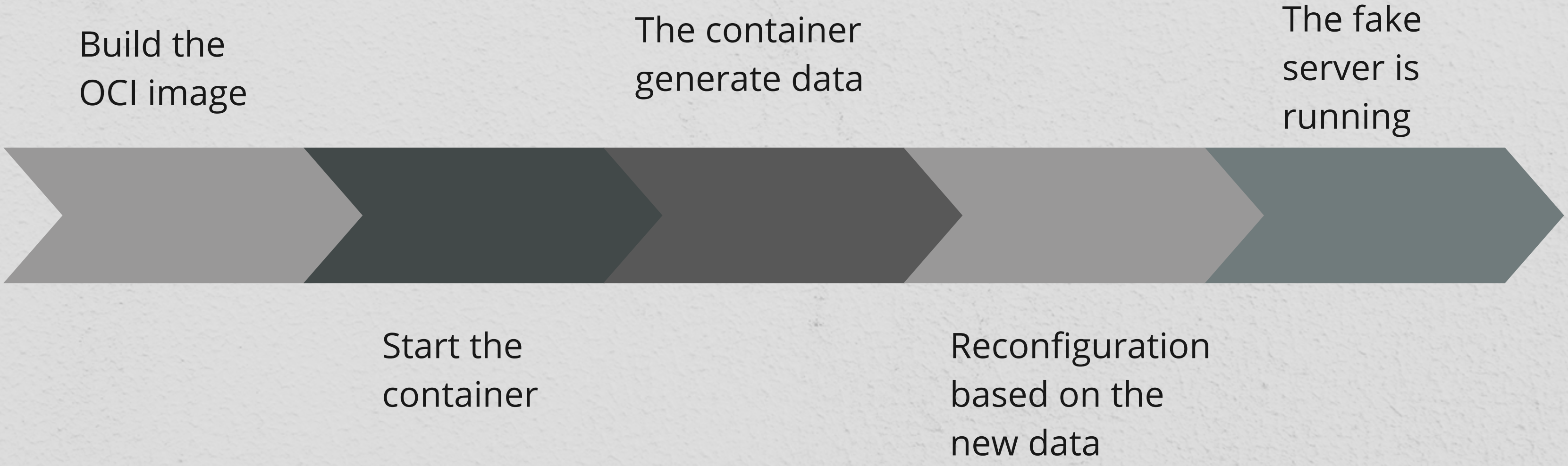
The Dockerfile

In the Dockerfile it's represented how the container is made



```
# Image
FROM debian:buster-backports
USER root

ENV DEBIAN_FRONTEND=noninteractive
ENV LDAP_DEBAUG_LEVEL=256

# Configuration variables

ENV DATA_DIR="/init/data"
ENV CONFIG_DIR="/init/config"

ENV LDAP_DOMAIN=example.com
ENV LDAP_ORGANISATION="Example, Inc"
ENV LDAP_BINDDN="cn=admin,dc=example,dc=com"
ENV LDAP_SECRET=admin

# Install and updates
RUN apt-get update && apt-get upgrade -y && apt-get install --no-install-recommends -y \
    wget build-essential libreadline-dev libncursesw5-dev libssl-dev libsqlite3-dev tk-dev libgdbm-dev
libc6-dev libbz2-dev libffi-dev zlib1g-dev\
    vim \
    slapd \
    ldap-utils \
    ldapscripts \
    systemctl \
    schema2ldif \
    curl \
    ca-certificates && \
    rm -rf /var/lib/apt/lists/*

# Ollama
RUN pip3.11 install --upgrade pip && pip3.11 install langchain
RUN curl https://ollama.ai/install.sh | sh

# Copy generated files to the container
COPY ./init /init

# Expose the LDAP port
EXPOSE 10389 10636

# Command to start ldap server
CMD ["/bin/bash", "/init/init.sh"]
```

The workflow



Build the OCI image

Start the container

The container generate data

Reconfiguration based on the new data

The fake server is running

03

DATA GENERATION

LLM for data generation

Pre-trained

Trained on billion of parameters

Easy to use

Download the model and make a script to interact

Suited for generation

They can generate data, based on an input that describe what you want

Two possible way of run LLMs locally

LLama-cpp-python

Python bindings for llama-cpp library

Ollama

Project to use LLM locally like containers

Which one?

Llama-cpp-python

- ✕ Output less heterogeneous
- ✕ Long time to execute
- ✕ Insert LLM model into the container to run it

Ollama

- ✕ Output more heterogeneous
- ✕ Shorter execution time
- ✕ Easy integration with docker
- ✕ Download the LLM model inside the container to run it

Final thoughts

Feasible improvments

- ✕ Make a lighter OCI image
- ✕ Avoid generate data inside the container to make it lighter and faster
- ✕ Use pre-generated data or a different machine to produce them

Resources

- Deception component
- OpenLDAP
- Docker
- Langchain
- Llama-cpp-python
- Ollama