LANTHORN SECURITY & PRIVACY DESIGN

SOLUTION TECHNICAL DOCUMENT

LANTHORN.AI

Affordable & Secure Occupancy Analytics

Privacy and Security

Summary

Lanthorn offers a product that ensures the security and privacy of your data..

A summary of the key points discussed is:

- **Local processing:** All the processing of the information is done in edge devices in your local network without exposure of the data to the cloud**.**
- **Local storage:** All the generated information is stored inside the device.
- **Keep the person's privacy**: All the generated data is anonymous. The processor only stores statistical information.
- **Secure public dashboard:** The public dashboard is built to avoid storing any critical information outside the local network. Only the **dashboard account information and the processor connection information** (url, user and password), the latter **encrypted by the frontend**, are stored in our servers.
- **Encrypted communication in API calls:** You can enable HTTPS and enforce encryption on requests.
- **OAuth2 authentication in API calls:** You can enforce authentication in your API calls.

Security in AI systems

The use of AI is becoming more popular in all industries. There is always a new product that uses machine learning. Usually, these products advertise their advantages in terms of ROI; however, at times there is uncertainty regarding the security aspects involved in their usage.

AI solutions need access to your data to develop their tasks. At times, the data that they access is private or confidential, which involves a risk of data breach. In order to mitigate security risks when evaluating AI solutions, you should ask your provider the following questions:

- Where is the data processed?
- What data is stored? Where is it stored?
- Can I run your solution in my network without any exposure to the internet?

At Lanthorn, we ensure the security of your data. None of your data will leave your servers unless you choose to export it. This article explains the security behind our solution to give you the necessary guarantees.

Lanthorn architecture

The architecture of the Lanthorn can be split into 2 components:

- Lanthorn processor
- Lanthorn dashboard

Lanthorn processor

The processor is the unit responsible for video processing and is designed to run efficiently on multiple edge devices. The list of supported devices includes:

- NVIDIA Jetson Nano
- NVIDIA Jetson TX2
- Coral Dev Board
- AMD64 node with attached Coral USB Accelerator
- X86 node
- X86 node accelerated with OpenVino toolkit
- X86 node with Nvidia GPU

The processor is the **only one component that consumes from the cameras' video streams**. To use it, you only need to clone the repository in one of the supported devices and configure it to consume the stream of any camera that you have connected to your local network (either in the .ini file or in the Lanthorn Dashboard). The steps to achieve this are documented both in the README on GitHub and at Lanthorn's Get Started Page.

Since the processor can be deployed in any of the above listed small, low-power edge devices, **you do not need to expose any of your camera streams outside of your local network (LAN)**; you only need to include the device inside the network. This configuration guarantees that the processing of the **information stays inside of your private network**; without transmitting anything to the cloud. Taking into account that the input of Lanthorn is videos, maintaining them inside the network is important to keep the privacy of the people recorded.

The result of the processor is a set of CSV **files** that are also **stored on the device**. There is no need to expose any of these files outside of the network.

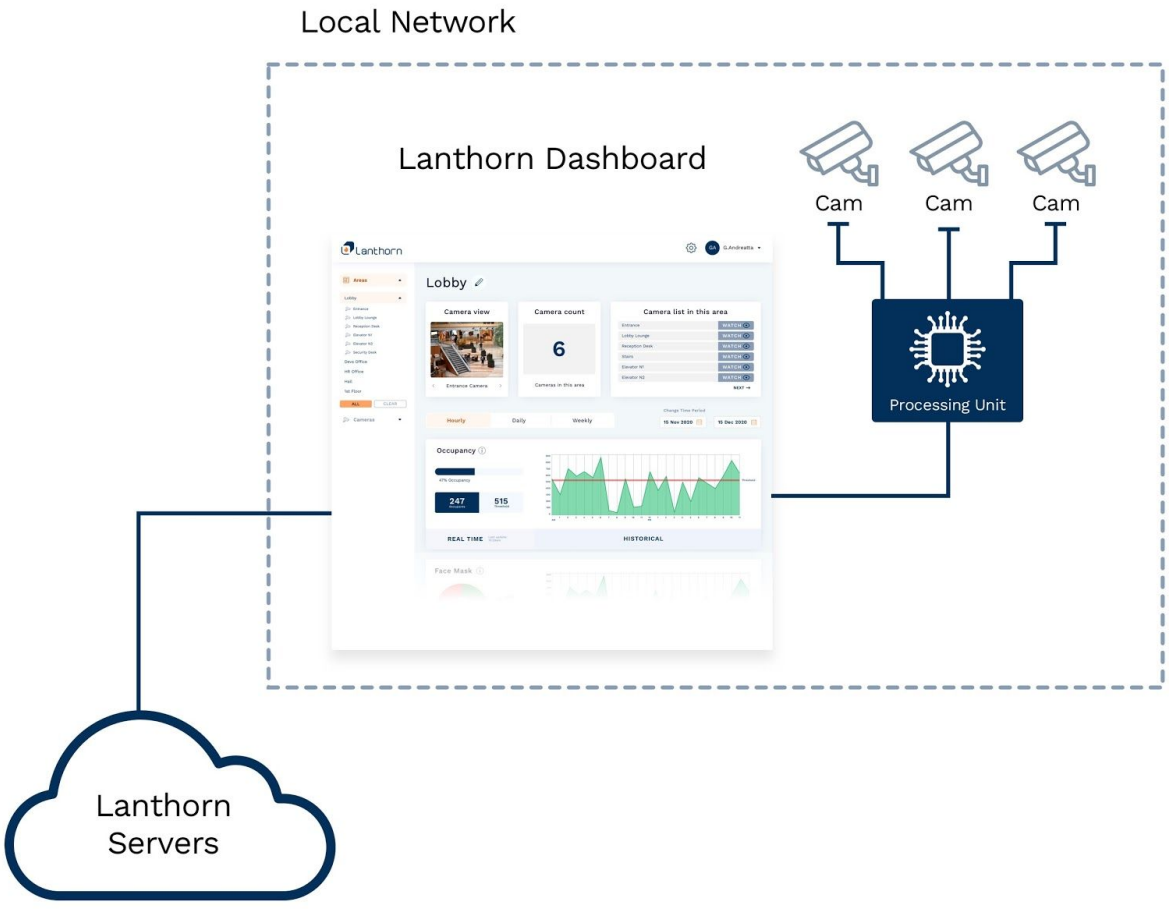
Lanthorn Dashboard

A public web application built in React and Python visualizes and analyzes the results generated by the processor. The subscription to the dashboard is **free**, you only need to sign up on app.lanthorn.ai. It is not mandatory to subscribe to app.lanthorn.ai to use the processor; you can analyze the data directly from the generated files, using the exposed endpoints, or by building your own frontend.

Nevertheless, you can use this public frontend by Lanthorn **without risking your private data** as the **only information stored** in that application is the **account information**; all other information (video, CSV files, etc) is **stored inside the processor**.

Once you log-in to the application, the first step that you need to complete is “Setting up the processor”**.** There you will be asked for the processor url, username and password (when Oauth is configured). These values will be **encrypted in the frontend using your password** as key and stored into the dashboard backend. The **encrypted url, user and password are the only values that we store**; the rest of the values are configured directly into the processor. After the setup, the whole **communication is between the browser and the processor** directly; this means that it is the **Web Browser’s client (on your PC**) **that communicates with your local processor**, instead of a Cloud Server that renders the web page.

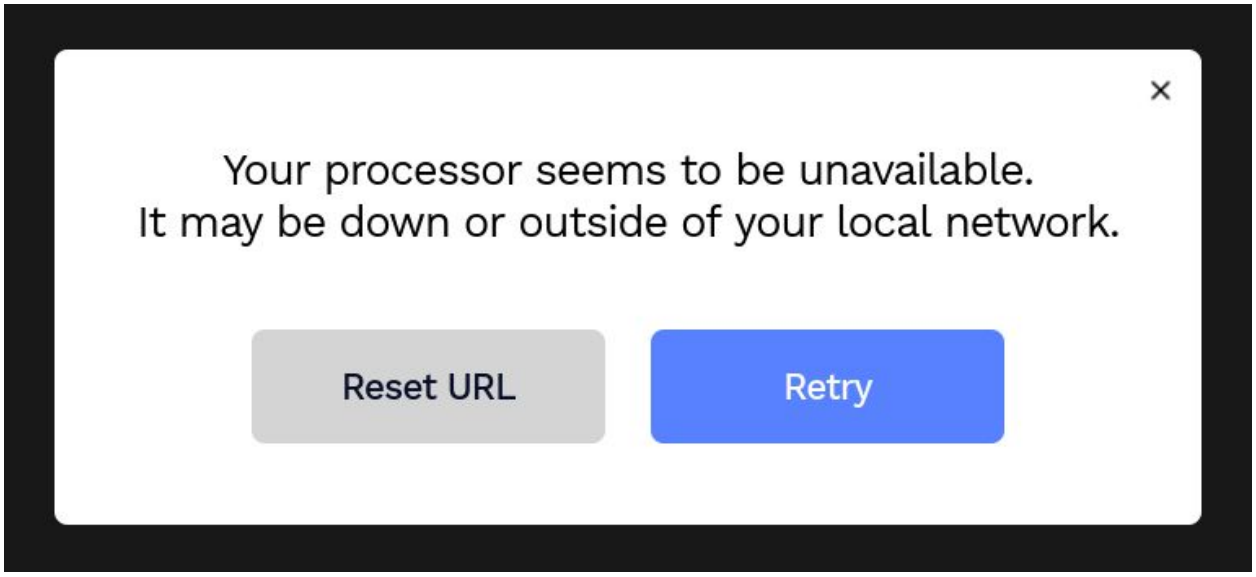
The following diagram shows the previously discussed architecture:



Processor API security

The way that a dashboard (and any other third-party application) interacts with the processor is through API endpoints (documented here). **You don’t need to expose these endpoints outside your network**, as

the browser interacts directly with the processor. If you try to log in into the dashboard outside the network, you will receive the following error:



Enabling HTTPS/SSL

You can **increase the security** in your API by **enabling HTTPS** in the processor, which means that the communication between the browser (or any client) and the processor will be **encrypted using SSL**.

If the processor is only exposed in your local network, you will not have any domain attached to it. Without a domain, none of the well-trusted certificate authorities (CA) will sign your certificate. So, before enabling SSL you will need to create a **self-signed certificate** (attached to your local IP) and import it into your browser. We recommend using the scripts mentioned here to create a local CA and a certificate for your local IP with openssl.

After enabling HTTPS in your processor, the **not-secure requests over HTTP will be rejected,** allowing only encrypted communication. We strongly recommend enabling HTTPS in the processor to ensure the privacy of your data. Not exposing the endpoints outside the local networks reduces risk. However, without HTTPS, any device inside the LAN can intercept the messages exchanged between the processor and read them.

If you want to use the dashboard, as it is published over HTTPS, you will need to follow the steps above in order to enable HTTPS in the processor. If you would rather communicate through HTTP, then you must edit your site settings for https://beta.lanthorn.ai in order to allow for Mixed Content (Insecure Content).

Without doing either of these steps the browser will block all the insecure requests to the processor.

Configuring OAuth2 in all the endpoints

Finally, you can enable OAuth2 authentication in all your endpoints. We recommend enabling it to force all the clients to send a valid JSON Web Token (JWT) as an **authorization header** in all the requests. Without doing it, any person with access to your LAN (and knowledge of the processors' IP) can request any of the exposed endpoints. Depending on the number of persons with this access, the vulnerability can be considered a risk or not.

To request a valid JWT you need to know the name and password of the configured user (as is explained here). The user name and **hashed password** are stored inside the processor and are used to generate a new valid token each time that someone requests a new one. If you are using the dashboard, these credentials will be also stored **encrypted** in our backend (same as the processor url).

Configuring OAuth2, you will ensure that the only way to interact with the processor API is by knowing the user and its password. Otherwise, **you will not be able to request** a valid JWT and all your requests will receive an **UNAUTHORIZED** response.

Personal privacy

At Lanthorn, one of our objectives is to **keep the privacy of your data**. We are aware that the processed videos probably contain sensitive information; such as the identity of the people recorded. For that reason, we build the processing pipeline taking it into account.
The videos are **processed and stored inside your local network** (as is explained in the "Lanthorn processor" section), avoiding any unwanted exposition outside your organization. Moreover, the **process doesn't require face recognition** to measure to accurately detect occupancy.
All the **data generated and stored is also anonymized to respect personal privacy**; storing only statistical information, such as the number of people. We **only store videos if you enabled** the live video feed results; however, all **faces are blurred** (no facial recognition)