Video Portal Security Design

Overview

- Synergyse Training for Google Apps™ video portal is a web-based solution that provides video training for Google Apps™.
- The purpose of this document is to provide a high level overview of video portal security design.

Security Architecture

- Frontend
  Our frontend service is a web portal accessible over the HTTPS protocol. The frontend makes requests to the backend on an internal network that is not exposed to the internet. It can be accessed by end-users when you deploy the Synergyse Marketplace application from the Google Apps Application Launcher menu. It can also be accessed through the link http://synergyse.com/videos

- Backend
  Our backend service is completely hosted on the Google Cloud Platform, and runs on Google App Engine, Google Cloud SQL & Google Cloud Storage. From a data security perspective, we are housed in the same data centers as Google Apps. All data is encrypted (128-bit AES) and secured (TLS) during transport and storage. We use Google Oauth 2.0 for authentication against Google accounts, which is Google's recommended method for securely authenticating.

- User Data
  Synergyse only has access to view your user's email addresses & organizational unit (via installation of the Synergyse Marketplace application). We do not have access to view any customer data, nor do we store any data in our database related to the user other than their email address & organizational unit. We send and store data only relating to usage of our training application, such as lesson playback and completion. This is used so you can generate reports of usage within your organization, and generate reports for specific organizational units.

- Custom Lesson Data
  Administrators have the ability to add custom video lessons to the Synergyse training system, the videos can be uploaded directly to our Google Cloud Platform servers where they are hosted privately and encrypted (128-bit AES). Custom video content that you upload to our system is hosted privately for your organization and accessed through secure authenticated (via Google Oauth) links in the frontend system.