Ready Room Security

Infrastructure

Ready Room runs on the Google Cloud Platform (GCP). By virtue of this we inherit all of Google's infrastructure security, including physical access, network management, server provisioning, VM isolation, and more. See here for detailed information: https://cloud.google.com/docs/security/infrastructure/design

Database

Ready Room uses Google Cloud SQL, a fully managed instance of the PostgreSQL relational database. Data that is housed in Google Cloud SQL is always encrypted, both in the datastore and in backups. In addition, communication between the application servers and the database is also encrypted. That is, all data is encrypted at rest and in transit.

Database access credentials are managed by Ready Room's PaaS provider, Gigalixir, and the password is a fully random UUID.

Command line access to the database is available, but authentication is via private key authentication, a key that is stored securely and available to only one person.

The database is backed up automatically every 24 hours. Backups are maintained for seven days.

File Storage

All customer uploaded files are stored in Google Cloud Storage (GCS), a fully managed, secure, and durable object store. All communication between clients and GCS is encrypted. Files are stored in "buckets" configured with a uniform access control setting of "Not Public." That is, they are only accessible to a small number of authenticated users.

In order to read and write files to GCS, clients use a combination of Cross Origin Resource Sharing (CORS) and cryptographically signed URLs. These URLs are signed on the Ready

Room servers using a securely managed private key that pairs with a public key retained by Google. Signed URLs have a lifetime of only 15 minutes.

Application Servers

Ready Room is developed using the Elixir programming language and the Phoenix Application Server. Phoenix is a “secure by default” environment that ensures adherence to OWASP web application security best practices. Thus, Ready Room has built in protection against cross site scripting and cross site request forgery attacks. Phoenix uses the Ecto database access library which has built-in mitigation against SQL injection attacks.

Ready Room’s source code is maintained at Github. The Git version control system tracks all modifications to the source code repository. All source code changes are carefully reviewed by other developers for completeness, intent, and compliance with good development practices.

Only one user has access to the “main branch” of the source code that is compiled into a deployable artifact. This same user is the only employee who can modify our production environment. However, a “break glass” process is in place that allows a backup engineer to update production in the unlikely event that the principal engineer is unavailable and an emergency deployment is necessary.

Before going into production, source code is subject to a suite of automated tests to mitigate the risk of regressions. It is then deployed to our staging environment, where it undergoes a series of manual tests to ensure that the code works as intended. Only then can it be merged with our master branch and deployed to production.

Authentication

Registration

All users must be registered in the system before they can access it. The registration process is kicked off when a user is “invited” to the system by a superuser or administrator; users cannot self-register. When a user is invited into the system, only the user’s email address is requested and stored. System invitations use randomly generated tokens. Only people in possession of the token can accept an invitation. The tokens are hashed (SHA256) before storing in the database and expire after seven days. What information is collected at registration time depends on whether single sign-on is enabled or not.

Single Sign-On

We offer support for single sign-on to our customers who desire it and use a supported identity provider, currently Microsoft Azure Active Directory, Okta, and Cisco Duo. Single sign-on can only be enabled on an account by a Synclinical superuser. Single sign-on (SSO) allows users to use their corporate identity to sign in to Ready Room and to continue to leverage their corporate security practices, such as multi-factor authentication and password complexity rules.

When a user employed by an SSO-enabled organization accepts their Ready Room registration invitation, they will be asked only for their time zone. Other user data (currently just the user's full name and email address) is supplied by the identity provider (IdP) on successful login.

The login screen is the same for both traditional users and SSO-enabled users. SSO users need to know to click the "Log in with your Corporate Credentials" button. On doing so, the standard OIDC authentication flow kicks in.

As part of the authentication flow, we will ask the IdP for the following scopes:

- Azure AD: openid, email, offline_access, and https://graph.microsoft.com/user.read
- Okta: openid, email, and profile
- Duo: openid, email, and profile

Ready Room conforms to the following SSO best practices:

- SSO-enabled users are not able to log in via the standard username/password form
- Users cannot change or reset their password
- Users cannot change their email address or name
- Sessions expire when the browser is closed or the user actively logs out
- To avoid session fixation attacks, the session ID is replaced and all session information is destroyed at each login.

Application-Based Authentication

If a customer cannot use single sign-on, then Ready Room manages user information and the authentication process.

As with SSO-enabled users, traditional users must accept their registration invitation before they can log in. At that time, user's will set their full name, password, and time zone.

Ready Room passwords must be at least 12 characters long, but enforce no other patterns. Passwords are hashed via the Bcrypt standard before being stored in the database, this ensures robustness over time (in the face of increased processing power) and mitigates against timing attacks during authentication. At no time is anyone, even the database administrator, able to see the plaintext password. Users can reset their own passwords. If a user has forgotten their password, they can start a password-reset flow that also uses tokens. Password reset tokens expire after 24 hours.

Authentication is maintained across requests using a single signed cookie. That cookie is removed when the browser is exited, that is, it’s a session cookie. In addition, the user can select “remember me,” and Ready Room will keep that user authenticated for 60 days.

Authorization

Ready Room is a multi-tenant system. Synclinical developers have worked carefully to ensure that a tenant’s data is not leaked across boundaries. Within a tenant, we have very few roles (authorization contexts), these are:

Super User: Can create an account. Currently only the two Synclinical founders have superuser privileges. There is no code path that can be exploited to grant superuser privileges.
Admin: Can create and manage users. Can create inspections. Can access all inspections.
Team Members: Can participate in inspections, manipulating tasks and uploading documents.
Inspectors: Can view inspection requests that have been “released” and optionally download documents.

Access to an inspection is only available to admins and to users who have been explicitly added to the inspection.

Inspector accounts are special. The username and password are randomly generated and inspectors do not have control over their accounts. Access is limited to viewing only the tasks that have been released.

Access to an inspection can be disabled and reenabled at any time by an administrator. Disabled users can still log in so as to access other inspections. Users can be removed from the account entirely if they should have no further access to Ready Room.

Super users have *no* special access or visibility into customer accounts unless they are invited into the account as an admin or team member.

Personas

Besides roles, Ready Room also incorporates the notion of “personas.” By design most personas do not limit the actions a user currently adopting that persona can take. Personas are typically used to signal other team members what function a user is performing at the moment, such as “scribe” or “subject matter expert.” However, there are two special personas that have a positive impact on security: observer and restricted SME.

Observer

A team member who has been given the observer persona by an administrator, can see everything about an inspection but change nothing. They can see all requests, attachments, chat, scribe notes, etc., but they are not allowed to alter the state of the inspection in any way. This persona is commonly used to allow visibility into the inspection, but without the risk of altering, destroying, or releasing sensitive information.

Restricted Subject Matter Expert

By default, subject matter experts (SMEs) can see and alter any aspect of an inspection. This is generally not desirable if the SME is either not an employee of the sponsor or has had limited training on the system. A Restricted SME on the other hand, can only access the requests that have been assigned to them and are currently incomplete. This reduces the risk of external parties gaining access to internal information and the risk of accidental modifications or release of inspection data.

Network Access

Ready Room uses TLS (SSL) communication only. If a user attempts to access Ready Room over HTTP (unencrypted), they will be automatically redirected to the HTTPS endpoint. In addition, Ready Room uses Strict Transport Security such that the browser will never again allow an HTTP request to Ready Room, thus mitigating the “coffee shop attack” (credential sniffing).

System Integrity

Ready Room does not use GCP directly. Instead, it leverages a Platform as a Service (Paas) called Gigalixir (similar to Heroku). Gigalixir rebuilds docker containers from scratch with each deploy. The containers are running Heroku’s latest Ubuntu stack, Heroku-20, which is widely used and derived from the latest Ubuntu Linux release with minimal packages installed. In addition, Ready Room developers actively leverage current Elixir and JavaScript libraries which are upgraded frequently. Together, these practices mitigate against CVEs and zero-day vulnerabilities that may be lingering in old libraries.

Gigalixir itself uses GCP’s support for Kubernetes and Docker for OS and application isolation, while layering on additional security of its own: https://gigalixir.readthedocs.io/en/latest/main.html#how-secure-is-gigalixir

Finally, Ready Room developers run a static analysis tool against the code before each deployment. This tool (https://github.com/nccgroup/sobelow) looks specifically for security flaws.

Currently, the only finding is that Ready Room does not set Content-Security Policy headers in the HTTP request. This is correct and will be addressed in the future. In the meantime, we deem the lack of these settings to be of low risk.

Briefings

Ready Room supports integrated video conferencing, aka Briefings, which allow inspection team members and (optionally) external parties to participate in audio/video discussions.

The private meeting link used for team members is inherently secure. To access a meeting you must have been invited, which means you also must be an active Ready Room user, a member of that inspection, and logged in.

The public link is, by necessity, open to everyone, just as meeting links from Zoom and Google are. Even so, we have taken some pains to keep bad actors out of Ready Room briefings, to wit:

- The meeting ID in the public link is unguessable. It is generated using a cryptographically strong pseudo-random number generator and contains 1024 (one septillion) possible permutations.
- Participants using the public link are required to enter their name before attempting to join.
- All non-hosts must “knock” and be actively let into the briefing by the host.
- If the attendee’s camera is on, their picture, as well as their name, will be displayed to the host when they knock.

Furthermore, it is not possible to bypass Ready Room and go straight to our video conferencing provider, Whereby. Our integration with Whereby is such that Ready Room meetings are only accessible via the readyroom.net domain. Finally, Whereby themselves take security very seriously, here is their writeup on how communication is secured:
https://whereby.helpscoutdocs.com/article/526-data-storage-and-security

Shared Storage Integration

Ready Room can be configured to upload released documents to Box (box.com) or a variety of Microsoft services; Teams, OneDrive, and SharePoint online. To do this, we leverage Microsoft’s and Box’s support for OAuth 2.0.

When an inspector (typically) shares a folder with a team member, that team member needs to initiate the OAuth flow, which consists of authenticating to Box/Microsoft and retrieving a token that allows access to the shared folder. This access token is effective for only 60 minutes. Ready Room will attempt to refresh stale tokens as needed. Tokens cannot be refreshed after

60 (Box) or 90 (Microsoft) days. If the refresh token expires, the user must reauthorize to the storage provider.

Inspectors and Team Members should be aware that both of these services make it trivially easy to make shared folders public without intending to, and should be careful to restrict access only to named users.

OpenAI Integration

Ready Room integrates with OpenAI in two places.

1. Ready Room will send the title of a request to OpenAI in order to generate an embedding for the text. The returned embedding is used to ascertain where in the TMF a file may be located. This request is made every time a user views a request and cannot be turned off.
2. When chatting with “Reggie,” Ready Room’s AI chatbot designed to answer questions concerning regulations and guidance in drug and device development. Messages are sent to OpenAI’s ChatGPT model when a user submits a message to Reggie. No other messages are sent. The Reggie feature is disabled by default. An admin must enable Reggie on a per inspection basis.

When communicating with OpenAI, no account or personal identifying information is sent. Company names, user names, email addresses, and the like are not sent to OpenAI, just the title of requests and chat messages sent to Reggie.

Ready Room communicates to OpenAi via its API. Unlike when accessing ChatGPT directly, when using the API, OpenAI does not use customer data to train future versions of its models. That is, all inputs sent to OpenAI by Ready Room generate an output and are forgotten immediately by OpenAI.