HOBBES

Security & Privacy Overview

Last Updated: July 30, 2025

Hobbes is designed for teams that work with sensitive data, complex workflows, and strict compliance requirements. From the ground up, we've built the platform with a focus on security, privacy, and operational transparency - giving our customers confidence in how their data is handled.

This document outlines our deployment model, data controls, and security practices.

Deployment & Integration

Hobbes supports two integration methods:

- **CDN injection**: Fast and easy to implement (2-line script)
- **NPM install**: Statically bundled, version-locked, and auditable

Both options provide full control over how and when Hobbes code is loaded. Hobbes does not require elevated permissions nor access internal application state unless explicitly configured.

Data Access & Privacy Controls

Hobbes is built to operate securely in environments with strict privacy boundaries. By default, it does not collect page content, tokens, or app state unless explicitly enabled.

Key privacy features include:

1. **Page-level exclusions:** The widget can be disabled on any route or surface (e.g., internal tools or admin views)
2. **DOM redaction:** Sensitive elements can be excluded from capture using attribute flags or selector-based rules

3. **No implicit access:** The widget does not access user content, authentication tokens, or APIs unless configured to do so
4. **Blinded recordings:** Sensitive text / images can be masked before leaving the browser with configuration options available
5. **Data retention & deletion:** Customer data is retained only as long as required to support product functionality. Deletion can be requested at any time, and retention periods can be customized.

Customer Data Isolation

Hobbes enforces strict tenant separation by default:

- **Dedicated storage buckets per customer**
- **Logical access boundaries at the infrastructure and application level**
- **Audit trails for all data collection and access operations**

Security & Compliance

We are currently undergoing **SOC 2 Type II** certification and follow security best practices in how we develop and operate the platform. Our approach includes:

- TLS encryption in transit and AES-256 encryption at rest
- Secure secrets management using AWS Secrets Manager
- Role-based access control and internal audit logging
- Isolated production and development environments
- Scoped access to customer data with clear internal review paths

---

Contact

For security reviews, procurement questions, or incident escalation, reach out to us at: **security@hihobbes.com**

This document represents our current security posture and is subject to continuous improvement.