Synthetic Users Application Security Controls

Version: 1.0

Effective Date: 15 February 2026

Owner: Security & Compliance Lead

Approved by: CTO

---

1. Purpose

This document describes application-level security controls implemented by Synthetic Users to protect data integrity, prevent information leakage, and ensure secure data handling within the platform.

---

2. Scope

This policy applies to all Synthetic Users web applications, APIs, and backend services.

---

3. Cache Control & Information Leakage Prevention

3.1 Cache-Control Headers

- All pages and API responses that contain or display personal information are served with `Cache-Control: no-store` headers to prevent browser and intermediary caching of sensitive data.
- Proxy caches and CDN caching rules are configured to exclude authenticated endpoints and any response containing user-specific or personal data.

3.2 Session Data Protection

- Session tokens are stored securely and are not exposed in URLs, logs, or cached responses.
- Authenticated session data is transmitted only over HTTPS (TLS 1.2+).

3.3 Sensitive Data in Responses

- API responses are scoped to return only the data the requesting user is authorized to access.
- Personal information is not included in error messages, logs, or debugging output.

---

4. Data Validation & Integrity

4.1 Input Validation

- All data structures are validated at ingress and egress using **Pydantic schemas**, enforcing strict type checking, field validation, and data structure verification.
- Invalid inputs are rejected with appropriate error responses before reaching business logic or data storage.

4.2 Output Validation

- Data returned from APIs and backend services is validated against Pydantic schemas at egress, ensuring completeness, accuracy, and correct typing.
- This prevents malformed or incomplete data from being transmitted to clients or downstream services.

4.3 Validation Coverage

Boundary	Validation Method	Scope
API Ingress	Pydantic schema validation	All incoming request payloads

API Egress	Pydantic schema validation	All outgoing response payloads
Database Writes	Schema enforcement	All data persisted to database
External Service Integration	Schema validation	All data sent to/received from subprocessors

4.4 Error Handling

- Validation failures return structured error responses without exposing internal implementation details.
- All validation errors are logged for monitoring and debugging.

---

5. Additional Application Security Controls

- **Input Sanitization:** All user inputs to AI agents and scripting environments are sanitized using strict allowlists and schema validators.
- **Output Encoding:** Outputs generated from AI processing are encoded or escaped to prevent injection attacks in rendered environments.
- **Audit Logging:** All data operations are logged with metadata (timestamp, user ID, operation type) enabling forensic tracking.

---

6. Review

This policy is reviewed annually or when significant changes to the application architecture occur.