

Injection Defense Boundary Matrix v1

XSS, SQL Injection and Input Validation

Outcome. Classify untrusted data by interpreter/sink boundary, select context-appropriate XSS and SQL defenses, and use syntactic/semantic validation only where it actually contributes to the bounded control.

Evidence gate. Accept only when every critical dimension has inspectable evidence, the negative probe behaves as predicted, and changed-context transfer preserves a residual limitation.

XSS, SQL Injection and Input Validation Evidence Path

01 XSS CONTEXT Authority: XSS prevention depends on where untrusted data is placed; safe sinks and context-appropriate encoding/sanitization matter more than a single generic transform. Boundary: No executable attack payload is required or produced by the lesson.	02 SQL CODE DATA Authority: Parameterized queries preserve the SQL code/data separation for values and are preferred over string concatenation of untrusted data. Boundary: The lesson does not claim parameter binding can parameterize arbitrary SQL identifiers such as table names.	03 VALIDATION ROLE Authority: Input validation should check syntactic and semantic business rules early, but it is not a substitute for context-specific XSS or SQL injection defenses. Boundary: Passing validation does not make data universally safe.
04 SAFE API Authority: Selecting an API or sink that keeps data separate from executable interpretation reduces injection risk and is preferable to fragile ad-hoc escaping. Boundary: The lesson does not define every framework-specific escaping API.	05 DEFENSE EVIDENCE Authority: Injection evidence must prove the intended data remains data at the relevant sink/interpreter boundary and that invalid business input is rejected for the right reason. Boundary: A green unit test suite is not proof that every application sink is safe.	06 EVIDENCE BOUNDARY Authority: Separate source facts, local observations, bounded inferences, and human decisions. Boundary: Unavailable production behavior remains NOT_OBSERVED or NEEDS_VERIFICATION.

Identity, decision, and claim boundary

Learner		Reviewer	
Case / attempt		Capture date	
Scenario / concern		Decision	ACCEPT / REPAIR / DEFER

Claim boundary. This is a bounded executable lab. Retain implementation, applicable runtime, browser, DOM, or tool observations, mutation, repair, transfer, and residual-limit evidence; one passing checker is not whole-system proof. Runtime and security claims require retained output from the supplied bounded fixture. Source facts and static readiness are not promoted into production observations.

Required record fields and evidence tests

A field is complete only when its value is bounded, traceable, and classified. Use UNKNOWN when the required observation is unavailable or contradictory.

Field	Question to close	Minimum evidence	Invalid promotion
input_field	What does input field mean in this case?	Bounded scenario fact, source, or observable	Inference can be relabeled as observation
trust_source	What does trust source mean in this case?	Bounded scenario fact, source, or observable	Inference can be relabeled as observation
business_validation	What does business validation mean in this case?	Bounded scenario fact, source, or observable	Inference can be relabeled as observation
destination_sink_or_interpreter	What does destination sink or interpreter mean in this case?	Bounded scenario fact, source, or observable	Inference can be relabeled as observation
primary_defense	What does primary defense mean in this case?	Bounded scenario fact, source, or observable	Inference can be relabeled as observation
secondary_defense	What does secondary defense mean in this case?	Bounded scenario fact, source, or observable	Inference can be relabeled as observation
positive_evidence	What evidence closes or revises positive evidence?	Smallest relevant observation or counterexample	A named tool or source is assumed to have been checked
negative_evidence	What evidence closes or revises negative evidence?	Smallest relevant observation or counterexample	A named tool or source is assumed to have been checked
residual_boundary	What does residual boundary mean in this case?	Bounded scenario fact, source, or observable	Inference can be relabeled as observation

Evidence and failure states

Evidence: OBSERVED / INFERRED / NOT_OBSERVED / NEEDS_VERIFICATION. Decisions: ACCEPT / REPAIR / DEFER. Failure layers: XSS CONTEXT / SQL CODE DATA / VALIDATION ROLE / SAFE API / DEFENSE EVIDENCE.

Decision and evidence table

English support projection from derived_text/tables/SEC-007-TBL1.example.en-US.json. This example demonstrates field relationships; it does not supply the learner's transfer or final decision evidence.

Field	Governed example value	Evidence question	Boundary check
input_field	display_label	What observation supports input field?	Is the conclusion no stronger than this value?
trust_source	synthetic user field	What observation supports trust source?	Is the conclusion no stronger than this value?
business_validation	length/category rule	What observation supports business validation?	Is the conclusion no stronger than this value?
destination_sink_or_interpreter	HTML text preview	What observation supports destination sink or interpreter?	Is the conclusion no stronger than this value?
primary_defense	safe text sink + context encoding	What observation supports primary defense?	Is the conclusion no stronger than this value?
secondary_defense	CSP/framework defaults	What observation supports secondary defense?	Is the conclusion no stronger than this value?
positive_evidence	benign marker remains data	What observation supports positive evidence?	Is the conclusion no stronger than this value?
negative_evidence	changed context requires reclassification	What observation supports negative evidence?	Is the conclusion no stronger than this value?
residual_boundary	other sinks are not proven	What observation supports residual boundary?	Is the conclusion no stronger than this value?
Source-package validation	ACCEPTED_IMPORTABLE; 0 blockers; 0 warnings · QA_ACCEPTED_ELIGIBLE		

Complete Injection Defense Boundary Matrix v1

Enter one bounded value per required field. Every material entry needs an evidence locator and a state. Use additional workbook rows when the artifact needs multiple concerns, responsibilities, resources, or path steps.

Required field	Learner value	Evidence locator	State / review note
input_field			OBSERVED / INFERRED / UNKNOWN
trust_source			OBSERVED / INFERRED / UNKNOWN
business_validation			OBSERVED / INFERRED / UNKNOWN
destination_sink_or_interpreter			OBSERVED / INFERRED / UNKNOWN
primary_defense			OBSERVED / INFERRED / UNKNOWN
secondary_defense			OBSERVED / INFERRED / UNKNOWN
positive_evidence			OBSERVED / INFERRED / UNKNOWN
negative_evidence			OBSERVED / INFERRED / UNKNOWN
residual_boundary			OBSERVED / INFERRED / UNKNOWN

Human closure

Decision	ACCEPT / REPAIR / DEFER	Reviewer	
Residual uncertainty		Consequence	
Smallest next evidence		Evidence owner / date	

Run → observe → mutate → repair → transfer

Run the bounded fixture and retain observable results. Predict each deliberate mutation before executing it, then prove the repair and changed-context transfer.

Case	Bounded concern or change	Provisional decision	Deciding evidence	Learner observation	Stable / unresolved boundary
SIMPLE	Synthetic profile preview + report filter	XSS prevention depends on where untrusted data is placed; safe sinks and context-appropriate encoding/sanitization matter more than a single generic.	The learner maps benign marker data into synthetic output contexts and chooses the documented safe handling for each context.	<div></div>	<div></div>
NEGATIVE XSS CONTEXT	Benign marker routed to a changed output context	Parameterized queries preserve the SQL code/data separation for values and are preferred over string concatenation of untrusted data.	A local query builder fixture compares a parameterized path with a deliberately invalid string-building design using only benign.	<div></div>	<div></div>
NEGATIVE SQL DESIGN	Local query design changed to string_building	Input validation should check syntactic and semantic business rules early, but it is not a substitute for context-specific XSS or SQL injection.	The evidence matrix records one syntactic and one semantic validation decision separately from sink/interpreter controls.	<div></div>	<div></div>
TRANSFER	Rich comments + report sort identifier	Selecting an API or sink that keeps data separate from executable interpretation reduces injection risk and is preferable to fragile ad-hoc escaping.	The learner identifies the safe API boundary for the synthetic output/database cases.	<div></div>	<div></div>

Reasoning sequence

Phase	Action	Required capture	Truth rule
Boundary	Declare and bound input field	Scenario fact, evidence locator, or explicit UNKNOWN	Runtime and security claims require retained output from the supplied bounded fixture. Source facts and static readiness are not promoted into production observations.
Boundary	Declare and bound trust source	Scenario fact, evidence locator, or explicit UNKNOWN	Runtime and security claims require retained output from the supplied bounded fixture. Source facts and static readiness are not promoted into production observations.
Boundary	Declare and bound business validation	Scenario fact, evidence locator, or explicit UNKNOWN	Runtime and security claims require retained output from the supplied bounded fixture. Source facts and static readiness are not promoted into production observations.
Stress test	Run the negative xss context case and retain its evidence	Prediction, changed assumption, stable boundary, next evidence	A plausible counterexample does not become an observed implementation fact
Stress test	Run the negative sql design case and retain its evidence	Prediction, changed assumption, stable boundary, next evidence	A plausible counterexample does not become an observed implementation fact
Stress test	Run the transfer case and retain its evidence	Prediction, changed assumption, stable boundary, next evidence	A plausible counterexample does not become an observed implementation fact

Guardrail. A local deterministic fixture does not prove production identity, authorization, confidentiality, integrity, availability, recovery, enforcement, or system-wide security.

Changed-context transfer

Rich comments + report sort identifier

Transfer item	Changed-context requirement	Learner evidence	Review check
Scenario	State the changed concern, resource, or responsibility boundary.	_____	Worked scenario not copied
Observed facts	Record only facts supplied or directly observed.	_____	Inference stays separate
Concern split	Create separate rows for independent responsibilities or paths.	_____	No page-wide shortcut
Classification	Choose only ACCEPT / REPAIR / DEFER or preserve UNKNOWN.	_____	Placement is justified
Failure probe	Apply one counterexample that tests an over-broad claim.	_____	No invented mechanism
Non-claims	State at least two limits plus the smallest next evidence.	_____	Residual and next evidence stated

Non-compensatory review

Critical dimension	PASS / REPAIR	Evidence locator	Repair or rationale
Scenario and responsibility boundary correctness	_____ _____	_____	_____
Observation, inference, and source traceability	_____ _____	_____	_____
Decision rationale and next-evidence quality	_____ _____	_____	_____
Failure diagnosis and overclaim resistance	_____ _____	_____	_____
Transfer, residual uncertainty, and non-goals	_____ _____	_____	_____

Canonical QA gate

Validation	ACCEPTED_IMPORTABLE; 0 blockers; 0 warnings	Disposition	QA_ACCEPTED_ELIGIBLE
Blocker / note	No canonical-source blocker recorded.	Clearance receipt	_____

Media decision. Deck and video are NOT_APPLICABLE_WITH_RATIONALE; the embedded visuals, editable record, worksheet, and learner-authored reasoning are the assessed support.