# How Web Form Security Prevents Intrusion

## Overview

Web form security reduces attack surface and stops automated and manual intrusions by validating input, authenticating users, protecting sessions, and sanitizing output.

## Input Validation & Sanitization

Server-side validation (required) and client-side checks (UX) prevent malformed or dangerous data. Sanitization removes/encodes characters that could trigger SQL injection or XSS. In this codebase, use prepared statements and parameterized queries to avoid SQL injection, and escape output when rendering.

## Authentication & Email Verification

Require verified accounts and enforce strong password handling. Email verification prevents automated account abuse. Implement account lockouts and rate-limiting on login and resend verification endpoints to stop brute force and enumeration.

## CSRF Protection

Use anti-CSRF tokens for state-changing forms (login, register, update). Tokens ensure requests originate from legitimate pages, preventing cross-site request forgery attacks.

## Session Management & Secure Cookies

Use server-side session checks, regenerate session IDs after privilege changes, and set cookies with HttpOnly and Secure flags. Validate session on sensitive operations and log out on suspicious activity.

## Rate Limiting & Logging

Apply rate limits to critical endpoints (login, resend verification) and log suspicious events for analysis. Logs help detect repeated intrusion attempts and support incident response.

## Summary

Combining validation, sanitization, authentication, CSRF protection, secure session handling, and monitoring creates layered defenses that prevent most common intrusion vectors in web forms.