Secure Code Analyzer Report

Generated at: 2025-08-31 11:51:42 UTC

	Snippet	Suggestion
	eval(req.query.code); // code injection	Avoid eval(); us
	child_process.exec('ls ' + req.query.dir); // command injection	Use safer spaw
	document.body.innerHTML = req.query.html; // XSS in client-side route (demo)	Use textConten
	const h = crypto.createHash('md5').update(req.query.q || 'x').digest('hex');	Use SHA-256/5
	const express = require('express');	Pass a function
	const crypto = require('crypto');	Pass a function
	const child_process = require('child_process');	Pass a function
	app.get('/hash', (req, res) => {	Pass a function
	app.get('/run', (req, res) => {	Pass a function
	res.send("ok");	Pass a function
	app.get('/eval', (req, res) => {	Pass a function
	res.send("done");	Pass a function
	app.get('/search', (req, res) => {	Pass a function
	res.send("searching");	Pass a function
	app.get('/xss', (req, res) => {	Pass a function
	res.send("done");	Pass a function
	console.log("Admin section visible!"); // BAD: client-side auth check	Remove consol
	console.log("Debug: reached end of app2.js");	Remove consol
	if (user.role === "admin") {	Do not rely on c
	const cspHeader = "Content-Security-Policy: default-src 'self' 'unsafe-inline'";	Avoid unsafe-inl
	const jqueryVer = "jquery-1.12.4.min.js";	Upgrade to lates
	const userUrl = "http://" + window.location.search.replace("?url=", "");	Pass a function
	document.getElementById("demo").innerHTML = window.location;	Use textConten
ution like eval(); | setInterval called with a string → dynamic code execution like eval()	setTimeout("alert('XSS')", 1000);	Use function ref
like eval(); | Sensitive data stored in localStorage/sessionStorage	localStorage.setItem("password", "12345");	Do not store se
	eval(code); // SINK	Avoid eval(); us
	eval(payload); // SINK	Avoid eval(); us
	container.innerHTML = '<h3>Search:</h3>' + untrusted1; // SINK	Use textConten
	a.innerHTML = '<a href="' + u + '">go</a>'; // SINK	Use textConten
	const token = Math.random().toString(36).slice(2); // SINK	Use crypto; get
	console.log('Fetched length (insecure):', (await res.text()).length);	Remove consol
	console.log('Weak token:', token);	Remove consol
	console.log('Polluted?', (check).pwned === true);	Remove consol
	console.log('Testing user regex length=', pattern.length);	Remove consol

	Snippet	Suggestion
	localStorage.setItem('authToken', t); // SINK	Do not store sec
	window.addEventListener('message', (ev) => {	Always validate
	const f = new Function('return (' + fnBody + ')'); // SINK	Avoid new Func
	if (later) setTimeout(later, 50); // SINK (string form)	Use function ref
	q: () => getParam('q'),	Pass a function
	json: () => getParam('json'),	Pass a function
	url: () => getParam('url'),	Pass a function
	html: () => getParam('html'),	Pass a function
	code: () => getParam('code'),	Pass a function
	target: () => getParam('target'),	Pass a function
	token: () => getParam('token'),	Pass a function
	re: () => getParam('re'),	Pass a function
	msg: () => getParam('msg')	Pass a function
	const container = document.getElementById('out1') || document.body;	Pass a function
	container.insertAdjacentHTML('beforeend', '<div class="result">' + untrusted2 +	Pass a function
	const btn = document.getElementById('dangerBtn') || document.createElement('butt	Pass a function
	btn.setAttribute('onclick', handler); // SINK	Pass a function
	if (typeof ev.data === 'string' && ev.data.startsWith('RUN:')) {	Pass a function
	const a = document.createElement('div');	Pass a function
	const el = document.createElement('div');	Pass a function
	el.setAttribute('style', v); // SINK	Pass a function
	await fetch('/api/echo', { headers: { 'X-Note': msg } }); // SINK (potential spl	Pass a function
	const c = document.getElementById('out2') || document.body;	Pass a function
	const link = document.createElement('a');	Pass a function
	if(isset($_GET['cmd'])){	Avoid shell exec
	echo $_GET['html']; // XSS	Encode output w
	$hash = md5($_GET['p']); // weak crypto	Use password_
	var_dump($e); // error leak	Log errors serve
	system("ls " . $_GET['cmd']); // command injection	Avoid system();
	eval($code); // ■ Dangerous	Avoid eval(); us
	$file = $_GET['file'];	Avoid shell exec
	$hash1 = md5($password); // ■ Weak hash	Use password_
	$hash2 = sha1($password); // ■ Weak hash	Use password_
	$result = mysqli_query($conn, $query);	Always use prep
	$page = $_GET['page'];	Never include u
	move_uploaded_file($_FILES['file']['tmp_name'], "uploads/" . $_FILES['file']['na	Validate file type
	system("cat " . $file); // ■ Dangerous	Avoid system();

	Snippet	Suggestion
	include($page . ".php"); // ■ Insecure include	Never use user
	$response = file_get_contents("http://" . $url);	Avoid using unv
	$data = file_get_contents($filename);	Avoid using unv
	if ($_GET['role'] === 'admin') {	Enforce server-s
	$secret = "mySuperSecretApiKey123";	Use environmer
	$conn = mysql_connect("localhost", "root", "password");	Migrate to PDO
	echo preg_replace("/.*/e", "system('ls')", $input);	Avoid shell exec
	$file = $_GET['page'];	Never include u
	$obj = unserialize($data);	Avoid unserializ
	include($file);	Never use user