# Most Common Salesforce Vulnerabilities in 2023

as Caught by CodeScan

autorabit

## Avoid Calling System.Debug() (APEX - Major)

Number of Times Caught by CodeScan: **3,537,895**

**Threat:**
The System.debug() method in Salesforce is used to write debugging information to the debug logs. In production code, careless use of this method can lead to leakage of sensitive information like Personally Identifiable Information (PII) or Protected Financial Information (PFI).

**Scenario:**
A batch process is written that logs detailed debug information for each processed record, including the record's data fields. An unauthorized user gains access to the debug logs, where they can see the customer's name, address, and financial information that was logged during the processing.

## Field Level Security Vulnerabilities (APEX - Critical)

Number of Times Caught by CodeScan: **1,962,084**

**Threat:**
In Salesforce, field-level security (FLS) provides a security layer that restricts user access at the field level. If FLS vulnerabilities exist, this means that certain users can access, view, or edit sensitive fields that they are not authorized to interact with.

**Scenario:**
A Salesforce object contains fields for social security numbers and financial details. A user with access to the object but without the correct field-level security permissions can access and view these sensitive fields.

## Class Variable Fields Should Not Have Public Accessibility (APEX - Blocker)

Number of Times Caught By CodeScan: **811,195**

**Threat:**
In Salesforce, variables with public accessibility can be accessed from anywhere within the application, not just the class in which they are declared. If these variables hold sensitive data and are not properly secured, an attacker can exploit this to access or modify sensitive data.

**Scenario:**
A public class variable is used to hold the key for an encryption algorithm used to encrypt sensitive data. An unauthorized class in the system accesses this public variable and uses the encryption key to decrypt sensitive data it otherwise wouldn't have access to.

## Unencoded Formulas In Script Tags XSS (Visualforce - Critical)

Number of Times Caught by CodeScan: **785,892**

**Threat:**
In Salesforce's Visualforce pages, unencoded formulas within script tags are susceptible to Cross-Site Scripting (XSS) attacks. An attacker can inject malicious scripts through these unencoded formulas, which are then executed when the page is rendered. This can lead to a range of harmful actions like session hijacking, sensitive data theft, and defacement of user interfaces.

**Scenario:**
An attacker manipulates the input used in a Visualforce page's unencoded formula within a script tag. The page then executes the attacker's script when a user loads the page. The script steals the user's session cookie, allowing the attacker to impersonate the user and access their data on Salesforce.

## Object Permissions Should Not Be Permissive (Salesforce Metadata - Critical)

Number of Times Caught By CodeScan: **741,432**

**Threat:**
In Salesforce, object permissions define the access level users have to view, create, edit, or delete records. If object permissions are set too permissively, users may gain access to data and operations that they are not authorized to view or perform. This can lead to unauthorized data access, leakage, or corruption.

**Scenario:**
An object containing sensitive customer information is configured with 'Modify All' permissions. A user with these permissions has gained total control over the object.

## Unescaped Value Could Cause XSS (Visualforce - Critical)

Number of Times Caught By CodeScan: **558,321**

**Threat:**
In Salesforce Visualforce pages, unescaped values can become a vector for Cross-Site Scripting (XSS) attacks. XSS vulnerabilities allow attackers to inject malicious scripts into web pages viewed by other users, leading to potential theft of session cookies, defacement of the website, or redirection to malicious sites.

**Scenario:**
A Visualforce page displays user-generated content without escaping the value. An attacker adds a malicious script to their content. When other users view the content, the script executes, stealing session cookies and allowing unauthorized access to sensitive data.

## Avoid HTML Comments (Visualforce - Critical)

**Threat:**
HTML comments in Salesforce Visualforce pages can unintentionally expose sensitive information or system details that could be used by an attacker to understand the system's internal workings. This information can be used to gain further understanding of the system's architecture and how to compromise it.

Number of Times Caught
By CodeScan: **436,203**

**Scenario:**
A developer includes HTML comments in a Visualforce page outlining the page's functionality and underlying business logic. An attacker viewing the page's source code finds these comments and gains an understanding of the system architecture, aiding them in crafting targeted attacks.

## Sharing Should Be Configured On Classes With DML (APEX - Critical)

**Threat:**
In Salesforce, Apex classes that execute DML statements without proper sharing rules can potentially bypass the standard Salesforce security model, including field-level security, org-wide default settings, role hierarchies, and sharing rules. If the class modifies data, this can lead to unauthorized data access or modification.

Number of Times Caught
By CodeScan: **355,251**

**Scenario:**
An Apex class without sharing rules defined executes a DML operation on a data object. A user triggers this class and it accesses and manipulates data beyond the user's permissions.

## Avoid Inline CSS Styles (Visualforce - Critical)

Number of Times Caught By CodeScan: **264,876**

**Threat:**
Inline CSS in Salesforce Visualforce pages can make the application more susceptible to attacks like XSS and content spoofing. As style attributes are not subject to the same security controls as other attributes, an attacker can potentially exploit inline styles to inject malicious content or scripts.

**Scenario:**
Inline CSS styles in a Visualforce page are manipulated by an attacker to modify the page's appearance, tricking users into clicking malicious links. This can lead to actions performed under the guise of the user.

## Avoid Catching Generic Exceptions (APEX - Critical)

Number of Times Caught By CodeScan: **216,057**

**Threat:**
Catching generic exceptions in Salesforce Apex code can lead to inappropriate handling of errors and potential security risks. If the exception details include sensitive information and are not properly handled, they could expose critical system or user information to an attacker.

**Scenario:**
An Apex class catches a generic exception during a DML operation, hiding the actual error. Unresolved, the underlying issue is not flagged for remediation and lies in waiting for exploitation.

## Avoid Untrusted/Unescaped Variables In DML Query (APEX - Critical)

Number of Times Caught By CodeScan: **189,751**

**Threat:**
Apex allows Database Manipulation Language (DML) operations for users to insert, update, or delete records in Salesforce. Untrusted or unescaped variables in Apex DML queries can result in a SOQL injection vulnerability, where an attacker can manipulate a SOQL query to expose or alter data they're not authorized to access. This can lead to substantial data breaches and pose severe security risks to the organization.

**Scenario:**
An Apex class uses an unescaped variable in a DML query. An attacker manages to manipulate the variable's value to alter the query's logic, allowing them to retrieve or modify data beyond their permissions.

## Deserializing JSON Is Security-Sensitive (APEX - Critical)

Number of Times Caught By CodeScan: **171,160**

**Threat:**
JSON (JavaScript Object Notation) is a popular data format used in Salesforce for data interchange between server and client, and between internal components. Apex provides JSON serialization and deserialization capabilities which are commonly used for processing JSON data, but can introduce severe security risks. Deserialization is the process of converting serialized data - data transformed into a format that can be stored or transmitted and then reconstructed later - back into its original format. If an attacker can supply malicious serialized objects to be deserialized, they can pull off arbitrary code execution or injection attacks.

**Scenario:**
An Apex class deserializes untrusted JSON input without proper validation. An attacker crafts a malicious JSON object that, when deserialized, triggers unwanted code execution, leading to system compromise.

# External Script And Style Resources Should Be Avoided (Visualforce - Critical)

Number of Times Caught By CodeScan: **152,756**

**Threat:**

Visualforce allows developers to build custom UIs hosted natively on the Lightning platform. External scripts or style resources in Visualforce can pose significant security risks. If these external resources are compromised or manipulated by an attacker, they can be used as vectors for attacks such as Cross-Site Scripting (XSS), data interception, or injection attacks. These attacks can be used to steal sensitive user data, alter user interface behaviors, or compromise system integrity within the Salesforce environment.

**Scenario:**

A Visualforce page includes an external script resource. An attacker compromises the external server hosting the script, altering it to include malicious code. When the page loads, the malicious script executes, stealing user data.