What is Jinja2 template injection?

Jinja2 template injection is server-side template injection (SSTI) in the Jinja2 engine used by Flask and other Python apps. It happens when user input is concatenated into the template source and rendered, so `{{ ... }}` expressions from the attacker execute on the server. From there an attacker reads the Flask `config` (which often holds secrets), walks Python objects, and reaches OS command execution. The fix is to pass user input as template data, never as template source.

HOW IT WORKS

01 How it works and example

The standard probe is a math expression only a template engine would evaluate: send `{{7*7}}`, and if the response contains 49, input is being rendered as a template. On Jinja2 an attacker then escalates from expression evaluation to code execution by reaching Python builtins through object attributes, for example `{{ config.__class__.__init__.__globals__['os'].popen('id').read() }}` or `{{ cycler.__init__.__globals__.os.popen('id').read() }}`. Simply rendering `{{ config }}` often dumps the Flask configuration, including `SECRET_KEY` and database credentials. Payloads are shown here for defensive testing.

SOURCES

[1] OWASP Web Security Testing Guide

[2] MITRE CWE-1336: Server-Side Template Injection

[3] Jinja2 sandbox documentation

[4] OWASP Top 10

Test your application for SSTI and 30 or more other classes.

securelayer7.net/learn/application-security/jinja2-template-injection

Open online