El Alto Modeling Security/Vulnerability Report

This table represents the known exploitable and non-exploitable vulnerabilities in third party packages used in the project.

Repository	Group	Problem Code	Effective /Ineffective	Resolvable by Project	Impact Analysis	Action
modeling-etsicatalog	Django	CVE-2019-14234	Ineffective	Yes	An issue was discovered in Django 1.11.x before 1.11.23, 2.1.x before 2.1.11, and 2.2.x before 2.2.4. Due to an error in shallow key transformation, key and index lookups for django.contrib. postgres.fields.JSONField, and key lookups for django.contrib.postgres.fields.HStoreField, were subject to SQL injection. This could, for example, be exploited via crafted use of "OR 1=1" in a key or index name to return all records, using a suitably crafted dictionary, with dictionary expansion, as the **kwargs passed to the QuerySet.filter() function.	Plan to update the no vulnerability version in F version by updating the version of Django
modeling-etsicatalog	Django	CVE-2019-14232	Ineffective There is no frondend which lets user to input.	Yes	An issue was discovered in Django 1.11.x before 1.11.23, 2.1.x before 2.1.11, and 2.2.x before 2.2.4. If django.utils.text.Truncator's chars() and words() methods were passed the html=True argument, they were extremely slow to evaluate certain inputs due to a catastrophic backtracking vulnerability in a regular expression. The chars() and words() methods are used to implement the truncatechars_html and truncatewords_html template filters, which were thus vulnerable. The `Django` package is vulnerable to Regular Expression Denial of Service (ReDoS) attacks via Catastrophic Backtracking. The `_truncate_html` function in `text.py` uses regular expressions prone to catastrophic backtracking, where parsing an input string with the expressions can result in constant re-evaluation due to the large number of possible variations, taking an exorbitant amount of time to complete, if at all. An attacker can exploit this via a maliciously crafted HTML input string that when parsed using the regular expressions will result in catastrophic backtracking and ultimately a Denial of Service (DoS) due to excessive resource consumption.	Plan to update the no vulnerability version in F version by updating the version of Django
modeling-etsicatalog	Django	CVE-2019-14233	Ineffective There is no frondend which lets user to input.	Yes	An issue was discovered in Django 1.11.x before 1.11.23, 2.1.x before 2.1.11, and 2.2.x before 2.2.4. Due to the behaviour of the underlying HTMLParser, django.utils.html.strip_tags would be extremely slow to evaluate certain inputs containing large sequences of nested incomplete HTML entities. The `Django` package is vulnerable to Uncontrolled Resource Consumption. The `strip_tags` function in `html.py` is very resource exhaustive when parsing large sequences of nested incomplete HTML entities. An attacker can exploit this by providing such an input and causing Uncontrolled Resource Consumption, eventually leading to a Denial of Service (DoS).	Plan to update the no vulnerability version in F version by updating the version of Django
modeling-etsicatalog	Django	CVE-2019-14235	Ineffective There is no frondend which lets user to input.	Yes	An issue was discovered in Django 1.11.x before 1.11.23, 2.1.x before 2.1.11, and 2.2.x before 2.2.4. If passed certain inputs, django.utils.encoding.uri_to_iri could lead to significant memory usage due to a recursion when repercent-encoding invalid UTF-8 octet sequences. Django is vulnerable to a Denial of Service (DoS) attack. The `repercent_broken_unicode()` function in the `encoding.py` file performs excessive recursion when attempting to percent-encode invalid UTF-8 octet sequences. A remote attacker can exploit this vulnerability by supplying a large number of invalid UTF-8 octet sequences via any affected input parameter. This will cause the application to consume a large amount of available resources, ultimately resulting in a DoS condition.	Plan to update the no vulnerability version in F version by updating the version of Django
modeling-etsicatalog	Django	N/A	Ineffective	Yes	The `Django` package is vulnerable to Cross-Site Scripting (XSS). The `global_settings.py` file sets the `SECURE_CONTENT_TYPE_NOSNIFF` security header to `False` by default, allowing browsers to sniff the content types of assets being fetched from a server. A remote attacker who can place input in a non-executable MIME type can exploit this behavior to trick a victim's browser into rendering the response as an executable MIME type with arbitrary content.	Plan to update the no vulnerability version in F version by updating the version of Django
modeling-etsicatalog	Django	N/A	Ineffective	Yes	Django - Cross-Site Request Forgery (CSRF) Project: https://docs.djangoproject.com/en/dev/releases/3.0/#security	Plan to update the no vulnerability version in F version by updating the version of Django

modeling-etsicatalog	Django	N/A	Ineffective	Yes	The `Django` package is vulnerable to Clickjacking. The `global_settings.py` file and the `get_xframe_options_value` function in `clickjacking.py` by default allowed the site to be framed by the same origin. An attacker who is able to generate content on the same origin as the vulnerable site could use this to social engineer users into interacting with functionality unintentionally.	Plan to update the no vulnerability version in F version by updating the version of Django
modeling-etsicatalog	Django	N/A	Ineffective etsicatalog main stream code does not use this function.	Yes	The qunit package is vulnerable to Cross-Site Scripting (XSS). The `testDone` function in `qunit.js` shows the source of tests in HTML. An attacker is able to provide a malicious unit test could exploit this behavior to execute arbitrary JavaScript in the browser of a victim who views the results of that unit test.	Plan to update the no vulnerability version in F version by updating the version of Django
modeling-etsicatalog	djangorest framework	N/A	Ineffective etsicatalog main stream code does not use `jquery` and bootstrap package.	Yes	etsicatalog main stream code does not use `jquery` and bootstrap package.	No action because this is the latest version of the package. Will continue to monitor in the F release.
modeling-etsicatalog	djangorest framework	CVE-2019-8331	Ineffective etsicatalog main stream code does not use `jquery` and bootstrap package.	Yes	etsicatalog main stream code does not use `jquery` and bootstrap package.	No action because this is the latest version of the package. Will continue to monitor in the F release.