

XML Processing Modules

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library)xml.rst, line 6)

Unknown directive type "module".

```
.. module:: xml
   :synopsis: Package containing XML processing modules
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library)xml.rst, line 9)

Unknown directive type "sectionauthor".

```
.. sectionauthor:: Christian Heimes <christian@python.org>
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library)xml.rst, line 10)

Unknown directive type "sectionauthor".

```
.. sectionauthor:: Georg Brandl <georg@python.org>
```

Source code: `:source:`Lib/xml/``

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library)xml.rst, line 12); [backlink](#)

Unknown interpreted text role "source".

Python's interfaces for processing XML are grouped in the `xml` package.

Warning

The XML modules are not secure against erroneous or maliciously constructed data. If you need to parse untrusted or unauthenticated data see the `:ref:`xml-vulnerabilities`` and `:ref:`defusedxml-package`` sections.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library)xml.rst, line 20); [backlink](#)

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library)xml.rst, line 20); [backlink](#)

Unknown interpreted text role "ref".

It is important to note that modules in the `:mod:`xml`` package require that there be at least one SAX-compliant XML parser available. The Expat parser is included with Python, so the `:mod:`xml.parsers.expat`` module will always be available.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library)xml.rst, line 25); [backlink](#)

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library)xml.rst, line 25); [backlink](#)

Unknown interpreted text role "mod".

The documentation for the `mod:xml.dom` and `mod:xml.sax` packages are the definition of the Python bindings for the DOM and SAX interfaces.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) xml.rst, line 30); [backlink](#)

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) xml.rst, line 30); [backlink](#)

Unknown interpreted text role "mod".

The XML handling submodules are:

- `mod:xml.etree.ElementTree`: the ElementTree API, a simple and lightweight XML processor

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) xml.rst, line 35); [backlink](#)

Unknown interpreted text role "mod".

- `mod:xml.dom`: the DOM API definition

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) xml.rst, line 40); [backlink](#)

Unknown interpreted text role "mod".

- `mod:xml.dom.minidom`: a minimal DOM implementation

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) xml.rst, line 41); [backlink](#)

Unknown interpreted text role "mod".

- `mod:xml.dom.pulldom`: support for building partial DOM trees

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) xml.rst, line 42); [backlink](#)

Unknown interpreted text role "mod".

- `mod:xml.sax`: SAX2 base classes and convenience functions

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) xml.rst, line 46); [backlink](#)

Unknown interpreted text role "mod".

- `mod:xml.parsers.expat`: the Expat parser binding

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) xml.rst, line 47); [backlink](#)

Unknown interpreted text role "mod".

XML vulnerabilities

The XML processing modules are not secure against maliciously constructed data. An attacker can abuse XML features to carry out denial of service attacks, access local files, generate network connections to other machines, or circumvent firewalls.

The following table gives an overview of the known attacks and whether the various modules are vulnerable to them.

kind	sax	etree	minidom	pulldom	xmlrpc
billion laughs	Vulnerable (1)	Vulnerable (1)	Vulnerable (1)	Vulnerable (1)	Vulnerable (1)
quadratic blowup	Vulnerable (1)	Vulnerable (1)	Vulnerable (1)	Vulnerable (1)	Vulnerable (1)
external entity expansion	Safe (5)	Safe (2)	Safe (3)	Safe (5)	Safe (4)
DTD retrieval	Safe (5)	Safe	Safe	Safe (5)	Safe
decompression bomb	Safe	Safe	Safe	Safe	Vulnerable

1. Expat 2.4.1 and newer is not vulnerable to the "billion laughs" and "quadratic blowup" vulnerabilities. Items still listed as vulnerable due to potential reliance on system-provided libraries. Check `:data:pyexpat.EXPAT_VERSION`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) xml.rst, line 73);
[backlink](#)

Unknown interpreted text role "data".

2. `mod:xml.etree.ElementTree` doesn't expand external entities and raises a `:exc:ParserError` when an entity occurs.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) xml.rst, line 77);
[backlink](#)

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) xml.rst, line 77);
[backlink](#)

Unknown interpreted text role "exc".

3. `mod:xml.dom.minidom` doesn't expand external entities and simply returns the unexpanded entity verbatim.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) xml.rst, line 79);
[backlink](#)

Unknown interpreted text role "mod".

4. `mod:xmlrpc.lib` doesn't expand external entities and omits them.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) xml.rst, line 81);
[backlink](#)

Unknown interpreted text role "mod".

5. Since Python 3.7.1, external general entities are no longer processed by default.

billion laughs / exponential entity expansion

The [Billion Laughs](#) attack -- also known as exponential entity expansion -- uses multiple levels of nested entities. Each entity refers to another entity several times, and the final entity definition contains a small string. The exponential expansion results in several gigabytes of text and consumes lots of memory and CPU time.

quadratic blowup entity expansion

A quadratic blowup attack is similar to a [Billion Laughs](#) attack; it abuses entity expansion, too. Instead of nested entities it repeats one large entity with a couple of thousand chars over and over again. The attack isn't as efficient as the exponential case but it avoids triggering parser countermeasures that forbid deeply-nested entities.

external entity expansion

Entity declarations can contain more than just text for replacement. They can also point to external resources or local files.

The XML parser accesses the resource and embeds the content into the XML document.

DTD retrieval

Some XML libraries like Python's `:mod:`xml.dom.pulldom`` retrieve document type definitions from remote or local locations. The feature has similar implications as the external entity expansion issue.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) xml.rst, line 106); [backlink](#)

Unknown interpreted text role "mod".

decompression bomb

Decompression bombs (aka [ZIP bomb](#)) apply to all XML libraries that can parse compressed XML streams such as gzipped HTTP streams or LZMA-compressed files. For an attacker it can reduce the amount of transmitted data by three magnitudes or more.

The documentation for [defusedxml](#) on PyPI has further information about all known attack vectors with examples and references.

The `:mod:`defusedxml`` Package

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) xml.rst, line 122); [backlink](#)

Unknown interpreted text role "mod".

[defusedxml](#) is a pure Python package with modified subclasses of all stdlib XML parsers that prevent any potentially malicious operation. Use of this package is recommended for any server code that parses untrusted XML data. The package also ships with example exploits and extended documentation on more XML exploits such as XPath injection.