20.4. XML Processing Modules

Source code: Lib/xml/

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 XML vulnerabilities and The defusedxml and defusedexpat Packages sections.

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

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

The XML handling submodules are:

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

- `xml.dom`: the DOM API definition
- `xml.dom.minidom`: a minimal DOM implementation
- `xml.dom.pulldom`: support for building partial DOM trees

- `xml.sax`: SAX2 base classes and convenience functions
- `xml.parsers.expat`: the Expat parser binding

20.4.1. 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**	**Vulnerable**	**Vulnerable**	**Vulnerable**	**Vulnerable**
quadratic blowup	**Vulnerable**	**Vulnerable**	**Vulnerable**	**Vulnerable**	**Vulnerable**
	Vulnerable	Safe (1)	Safe (2)	**Vulnerable**	Safe (3)

kind	sax	etree	minidom	pulldom	xmlrpc
external entity expansion					
DTD retrieval	**Vulnerable**	Safe	Safe	**Vulnerable**	Safe
decompression bomb	Safe	Safe	Safe	Safe	**Vulnerable**

1. `xml.etree.ElementTree` doesn't expand external entities and raises a `ParserError` when an entity occurs.
2. `xml.dom.minidom` doesn't expand external entities and simply returns the unexpanded entity verbatim.
3. `xmlrpclib` doesn't expand external entities and omits them.

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 `xml.dom.pulldom` retrieve document type definitions from remote or local locations. The feature has similar implications as the external entity expansion issue.

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.

20.4.2. The `defusedxml` and `defusedexpat` Packages

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.

defusedexpat provides a modified libexpat and a patched `pyexpat` module that have countermeasures against entity expansion DoS attacks. The `defusedexpat` module still allows a sane and configurable amount of entity expansions. The modifications may be included in some future release of Python, but will not be included in any bugfix releases of Python because they break backward compatibility.