Issue 11219

Title	A user can read the content of files on the machine running trytond by exploiting XEE vulnerability in camt54 parsing
Priority	critical
Status	resolved
Nosy list	ced, jeremy.mousset, nicoe, pokoli, reviewbot, roundup-bot, yangoon
Assigned to	ced
Keywords	review

Created on **2022-02-04.16:31:30** by **jeremy.mousset**, last changed **9 months ago** by **roundup-bot**.

Files

File name	Uploaded	Type	Details
screenshot_XEE.png	jeremy.mousset, 2022-02-04.16:32:24	image/png	view
bad_camt.xml	jeremy.mousset, 2022-02-04.16:31:30	text/xml	view

Messages

New changeset 69e8f18f5682 by Cédric Krier in branch 'default':
Protect against XML vulnerabilities
https://hg.tryton.org/tryton-env/rev/69e8f18f5682

New changeset a64ca55b86e3 by Cédric Krier in branch 'default':
Protect against XML vulnerabilities
https://hg.tryton.org/trytond/rev/a64ca55b86e3

New changeset 2ef295408a0b by Cédric Krier in branch '6.2':
Protect against XML vulnerabilities
https://hg.tryton.org/trytond/rev/2ef295408a0b

New changeset b8e700d01652 by Cédric Krier in branch '6.0':
Protect against XML vulnerabilities
https://hg.tryton.org/trytond/rev/b8e700d01652

New changeset d0744bba5682 by Cédric Krier in branch '5.0':
Protect against XML vulnerabilities
https://hg.tryton.org/trytond/rev/d0744bba5682

New changeset f801a89c84e7 by Cédric Krier in branch 'default':
Protect against XML vulnerabilities
https://hg.tryton.org/proteus/rev/f801a89c84e7

New changeset e4019b6ca238 by Cédric Krier in branch '6.2':
Protect against XML vulnerabilities
https://hg.tryton.org/proteus/rev/e4019b6ca238

New changeset 973635df5e61 by Cédric Krier in branch '6.0':
Protect against XML vulnerabilities
https://hg.tryton.org/proteus/rev/973635df5e61

New changeset 85eb95f609cf by Cédric Krier in branch '5.0':
Protect against XML vulnerabilities
https://hg.tryton.org/proteus/rev/85eb95f609cf

msg74307 (view) Author: [hidden] (ced) Date: 2022-02-22.22:54:54

I propose the March 1st to make the security release (with exiting bugfix) and the March 2nd to make the announce.

msg74002 (view) Author: [hidden] (jeremy.mousset) Date: 2022-02-10.17:28:21

It looks like xmlrpc is safe for this kind of attacks : https://docs.python.org/3.8/library/xml.html#xml-vulnerabilities

msg73888 (view) Author: [hidden] (ced) Date: 2022-02-04.17:03:51

Indeed for the `account_payment_sepa` (and others) which are parsing external XML with lxml, we should follow https://lxml.de/FAQ.html#how-do-i-use-lxml-safely-as-a-web-service-endpoint.
For `trytond` and `tryton` which uses `xmlrpc.client` which should use `defusedxml` if it is available. I do not think we must make it required because for the server the sanitization of the XML-RPC could be done by a proxy and `defusedxml` is slow as it is pure Python. Also for `tryton` it is less a problem as it receives only XML from normally a trusted server.

msg73887 (view) Author: [hidden] (jeremy.mousset) Date: 2022-02-04.16:32:24

I also attach a screenshot showing the issue of the client displaying the content of a file that is should not be served.

msg73886 (view) Author: [hidden] (jeremy.mousset) Date: 2022-02-04.16:31:30

How to reproduce the exploit:

On a trytond 6.2 server, (with lmxl==4.7.1) Create an incoming sepa message, fill the message field with the attached "bad_camt.xml" file (a file which conta
with references to the server local file "/etc/group"), and then "do" the message.

Result: The tryton client displays an error message containing the content of the /etc/group file

The trace of the error in the server logs is:

```
[...]
  File "/home/jeremy.mousset/.pyenv/versions/vanilla/lib/python3.8/site-packages/trytond/modules/account_payment_sepa/sepa_handler.py", line 38
    payments = self.get_payments(element)
  File "/home/jeremy.mousset/.pyenv/versions/vanilla/lib/python3.8/site-packages/trytond/modules/account_payment_sepa/sepa_handler.py", line 77
    ('kind', '=', self.get_payment_kind(element)),
  File "/home/jeremy.mousset/.pyenv/versions/vanilla/lib/python3.8/site-packages/trytond/modules/account_payment_sepa/sepa_handler.py", line 51
    return self._kinds[
KeyError: 'root:x:
[etc etc]
```

Of course the /etc/group file is only an example.

This issue belongs to the family of issues described here : https://cwe.mitre.org/data/definitions/611.html

How I found out:
we ran smegrep (https://semgrep.dev/) on our code base who reported the following:

> "Found use of the native Python XML libraries, which is vulnerable to XML external entity (XXE) attacks. The Python documentation recommends the 'defu
> instead. Use 'defusedxml'. See https://github.com/tiran/defusedxml for more information.

I then looked for a possibility of exploitation where etree.fromstring is used in the code.

I suppose there might be other ways to exploit xml parsing in trytond and its modules.
I don't know if the indication from the semgrep report to use https://github.com/tiran/defusedxml instead of the lxml library is the right one. It could be.

[I'm not sure if "critical" is the right priority here, but it is a serious issue in my opinion]

History

Date	User	Action	Args
2022-03-01 19:58:27	roundup-bot	set	messages: + msg74387
2022-03-01 19:57:55	roundup-bot	set	messages: + msg74385
2022-03-01 19:57:11	roundup-bot	set	messages: + msg74383 status: testing -> resolved
2022-02-22 22:54:54	ced	set	messages: + msg74307
2022-02-11 13:29:51	ced	set	component: + tryton, trytond, dashboard, account_payment_sepa, currency_ro, currency_rs
2022-02-10 20:50:27	ced	set	assignedto: ced keyword: + review reviews: 381941002 status: chatting -> testing
2022-02-10 17:28:21	jeremy.mousset	set	messages: + msg74002
2022-02-04 17:03:51	ced	set	messages: + msg73888 status: unread -> chatting
2022-02-04 16:38:55	jeremy.mousset	set	title: A user can read the content of files on the machine serving trytond by exploiting XEE vulnerability in camt54 parsing -> A user can read the content of files on the machine running trytond by exploiting XEE vulnerability in camt54 parsing
2022-02-04 16:32:24	jeremy.mousset	set	files: + screenshot_XEE.png messages: + msg73887

Showing 10 items. Show all history (warning: this could be VERY long)