Redmine - Defect #38539

Update Nokogiri to 1.15.2 in 5.0-stable and 4.2-stable

2023-05-11 10:17 - A Fora

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Go MAEDA	**% Done:**	0%
Category:	Security	**Estimated time:**	0.00 hour
Target version:	4.2.11		
Resolution:	Fixed	**Affected version:**	5.0.4

Description

Here's the details:

```
Name: activesupport
Version: 6.1.7.2
CVE: CVE-2023-28120
GHSA: GHSA-pj73-v5mw-pm9j
Criticality: Unknown
URL: https://discuss.rubyonrails.org/t/cve-2023-28120-possible-xss-security-vulnerability-in-safeb
uffer-bytesplice/82469
Title: Possible XSS Security Vulnerability in SafeBuffer#bytesplice
Solution: upgrade to '~> 6.1.7, >= 6.1.7.3', '>= 7.0.4.3'

Name: nokogiri
Version: 1.13.10
GHSA: GHSA-pxvg-2qj5-37jq
Criticality: Unknown
URL: https://github.com/sparklemotion/nokogiri/security/advisories/GHSA-pxvg-2qj5-37jq
Title: Update packaged libxml2 to v2.10.4 to resolve multiple CVEs
Solution: upgrade to '>= 1.14.3'

Vulnerabilities found!
```

Related issues:

Related to Redmine - Patch #38181: Update Nokogiri to 1.15.2	**Closed**
Related to Redmine - Patch #38374: Update Rails to 6.1.7.6	**Closed**

Associated revisions

Revision 22259 - 2023-06-22 03:40 - Go MAEDA

Update Nokogiri to 1.15.2 in 5.0-stable (#38539).

Patch by Holger Just.

Revision 22260 - 2023-06-22 03:42 - Go MAEDA

Update Nokogiri to 1.15.2 in 4.2-stable (#38539).

Patch by Holger Just.

History

#1 - 2023-05-16 18:22 - Holger Just

- Related to Patch #38181: Update Nokogiri to 1.15.2 added

#2 - 2023-05-16 18:25 - Holger Just

- Related to Patch #38374: Update Rails to 6.1.7.6 added

#3 - 2023-05-16 18:42 - Holger Just

The Rails vulnerability does (very likely) not affect Redmine 5.0. To quote the announcement:

> Ruby 3.2 introduced a new bytesplice method which ActiveSupport did not yet understand to be a mutation. Users on older versions of Ruby are

likely unaffected.

All users running an affected release and using bytesplice should either upgrade or use one of the workarounds immediately.

Redmine 5.0.x does not support Ruby 3.2. As such, it is (very likely) not affected by this issue. Still, we have updated the Rails version with #38374. This change will be released with Redmine 5.0.6.

The case with Nokogiri is a bit more complex. With Redmine, we support several older Ruby versions for which there are no Nokogiri releases anymore. This results in more complex version dependencies.

Yet, with Redmine 5.0, we are currently still pinning Nokogiri to ~> 1.13.10 for newer Rubies. This must be adapted to use ~> 1.14.3 for Ruby >= 2.7.0 only as nokogiri ended support for Ruby 2.6 with their 1.14.0 release. This can be fixed by adapting the version selection for nokogiri in the Gemfile.

For Redmine 4.2, this should be

```
gem 'nokogiri', (if Gem.ruby_version < Gem::Version.new('2.5.0')
                   '~> 1.10.10'
                 elsif Gem.ruby_version < Gem::Version.new('2.6.0')
                   '~> 1.12.5'
                 elsif Gem.ruby_version < Gem::Version.new('2.7.0')
                   '~> 1.13.10'
                 else
                   '~> 1.15.2'
                 end)
```

For Redmine 5.0, we can skip the first check as Redmine 5.0 supports only Ruby >= 2.5.0:

```
gem 'nokogiri', (if Gem.ruby_version < Gem::Version.new('2.6.0')
                   '~> 1.12.5'
                 elsif Gem.ruby_version < Gem::Version.new('2.7.0')
                   '~> 1.13.10'
                 else
                   '~> 1.15.2'
                 end)
```

For the trunk, we support only Ruby >= 2.7, so we can just use

```
gem 'nokogiri', '~> 1.15.2'
```

#4 - 2023-05-16 19:06 - Holger Just

- Status changed from New to Confirmed

#5 - 2023-06-15 14:58 - Holger Just

- Assignee set to Go MAEDA

Maeda-san, could you have a look at the changes in #note-3? Note that just edited my comment with the current nokogiri versions.

As for regularly updating the nokogiri versions, we may still want to consider just relaxing the dependency at least in trunk to something like '~> 1.15' so that any new versions are automatically used on bundle update. As nokogori marks their gem versions with supported Ruby versions, all bundler versions used with Ruby > 2.7 should be able to find the most recent supported version on their own. See #37100 for a more refined proposal regarding dependency updates.

#6 - 2023-06-21 08:57 - Go MAEDA

- Subject changed from Ruby vulnerabilities reported for v.5.0.5 (I cant select 5.0.5 from versions list) to Update Nokogiri to 1.15.2

- Target version set to 4.2.11

#7 - 2023-06-22 03:43 - Go MAEDA

- Subject changed from Update Nokogiri to 1.15.2 to Update Nokogiri to 1.15.2 in 5.0-stable and 4.2-stable

- Status changed from Confirmed to Closed

- Resolution set to Fixed

Committed the patch in #note-5 in r22259 and r22260. Thank you.