FastRuby
Audit Tool

42 Vulnerabilities found on your file

Advisories

42

VULNERABLE GEM: ACTIVESUPPORT@4.2.9

Name:
activesupport

Version:
4.2.9

ID:
CVE-2023-28120

LINK

Possible XSS Security Vulnerability in SafeBuffer#bytesplice

DESCRIPTION:

There is a vulnerability in ActiveSupport if the new bytesplice method is called on a SafeBuffer with untrusted user input. This vulnerability has been assigned the CVE identifier CVE-2023-28120.

Versions Affected: All. Not affected: None Fixed Versions: 7.0.4.3, 6.1.7.3

Impact

ActiveSupport uses the SafeBuffer string subclass to tag strings as html*safe after they have been sanitized. When these strings are mutated, the tag is should be removed to mark them as no longer being html*safe.

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.

Workarounds

Avoid calling bytesplice on a SafeBuffer (html_safe) string with untrusted

user input.

VULNERABLE GEM: ACTIVESUPPORT@4.2.9

Name:
activesupport

Version:
4.2.9

ID:
CVE-2023-22796

LINK

ReDoS based DoS vulnerability in Active Supportâ€™s underscore

DESCRIPTION:

There is a possible regular expression based DoS vulnerability in Active Support. This vulnerability has been assigned the CVE identifier CVE-2023-22796.

Versions Affected: All Not affected: None Fixed Versions: 5.2.8.15 (Rails LTS), 6.1.7.1, 7.0.4.1

Impact

A specially crafted string passed to the underscore method can cause the regular expression engine to enter a state of catastrophic backtracking. This can cause the process to use large amounts of CPU and memory, leading to a possible DoS vulnerability.

This affects String#underscore, ActiveSupport::Inflector.underscore, String#titleize, and any other methods using these.

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

Workarounds

There are no feasible workarounds for this issue.

Users on Ruby 3.2.0 or greater may be able to reduce the impact by configuring Regexp.timeout.

VULNERABLE GEM: ACTIVESUPPORT@4.2.9

Name:
activesupport

Version:
4.2.9

ID:
CVE-2020-8165

LINK

Potentially unintended unmarshalling of user-provided objects in MemCacheStore and RedisCacheStore

DESCRIPTION:

There is potentially unexpected behaviour in the MemCacheStore and RedisCacheStore where, when untrusted user input is written to the cache store using the `raw: true` parameter, re-reading the result from the cache can evaluate the user input as a Marshalled object instead of plain text. Vulnerable code looks like:

```
data = cache.fetch("demo", raw: true) { untrusted_string }
```

Versions Affected: rails < 5.2.5, rails < 6.0.4 Not affected: Applications not using MemCacheStore or RedisCacheStore. Applications that do not use the `raw` option when storing untrusted user input. Fixed Versions: rails >= 5.2.4.3, rails >= 6.0.3.1

IMPACT

Unmarshalling of untrusted user input can have impact up to and including RCE. At a minimum, this vulnerability allows an attacker to inject untrusted Ruby objects into a web application.

In addition to upgrading to the latest versions of Rails, developers should ensure that whenever they are calling `Rails.cache.fetch` they are using consistent values of the `raw` parameter for both reading and writing, especially in the case of the RedisCacheStore which does not, prior to these changes, detect if data was serialized using the raw option upon deserialization.

WORKAROUNDS

It is recommended that application developers apply the suggested patch or upgrade to the latest release as soon as possible. If this is not possible, we recommend ensuring that all user-provided strings cached using the `raw` argument should be double-checked to ensure that they conform to the expected format.

VULNERABLE GEM: ADDRESSABLE@2.5.2

Name:
addressable

Version:
2.5.2

ID:
CVE-2021-32740

LINK

Regular Expression Denial of Service in Addressable templates

DESCRIPTION:

Within the URI template implementation in Addressable, a maliciously crafted template may result in uncontrolled resource consumption, leading to denial of service when matched against a URI. In typical usage, templates would not normally be read from untrusted user input, but nonetheless, no previous security advisory for Addressable has cautioned against doing this. Users of the parsing capabilities in Addressable but not the URI template capabilities are unaffected.

VULNERABLE GEM: COMMONMARKER@0.17.9

Name:
commonmarker

Version:
0.17.9

ID:
GHSA-636f-xm5j-pj9m

LINK

Several quadratic complexity bugs may lead to denial of service in Commonmarker

DESCRIPTION:

IMPACT

Several quadratic complexity bugs in commonmarker's underlying `cmark-gfm` library may lead to unbounded resource exhaustion and subsequent denial of service.

The following vulnerabilities were addressed:

CVE-2023-22483 CVE-2023-22484 CVE-2023-22485 CVE-2023-22486

For more information, consult the release notes for version 0.23.0.gfm.7 .

MITIGATION

Users are advised to upgrade to commonmarker version 0.23.7 .

VULNERABLE GEM: COMMONMARKER@0.17.9

Name:
commonmarker

Version:
0.17.9

ID:
GHSA-48wp-p9qv-4j64

LINK

Commonmarker vulnerable to to several quadratic complexity bugs that may lead to denial of service

DESCRIPTION:

IMPACT

Several quadratic complexity bugs in commonmarker's underlying cmark-gfm library may lead to unbounded resource exhaustion and subsequent denial of service.

The following vulnerabilities were addressed: * CVE-2023-24824 * CVE-2023-26485

For more information, consult the release notes for versions 0.23.0.gfm.10 and 0.23.0.gfm.11.

MITIGATION

Users are advised to upgrade to commonmarker version 0.23.9

VULNERABLE GEM: COMMONMARKER@0.17.9

Name:
commonmarker

Version:
0.17.9

ID:
CVE-2024-22051

LINK

Integer overflow in cmark-gfm table parsing extension leads to heap memory corruption

DESCRIPTION:

IMPACT

CommonMarker uses `cmark-gfm` for rendering Github Flavored Markdown. An integer overflow in `cmark-gfm` 's table row parsing may lead to heap memory corruption when parsing tables who's marker rows contain more than UINT16_MAX columns. The impact of this heap corruption ranges from Information Leak to Arbitrary Code Execution.
If affected versions of CommonMarker are used for rendering remote user controlled markdown, this vulnerability may lead to Remote Code Execution (RCE).

PATCHES

This vulnerability has been patched in the following CommonMarker release:

v0.23.4

WORKAROUNDS

The vulnerability exists in the table markdown extensions of `cmark-gfm`. Disabling any use of the table extension will prevent this vulnerability from being triggered.

REFERENCES

https://github.com/github/cmark-gfm/security/advisories/GHSA-mc3g-88wq-6f4x

ACKNOWLEDGEMENTS

We would like to thank Felix Wilhelm of Google's Project Zero for reporting this vulnerability

FOR MORE INFORMATION

If you have any questions or comments about this advisory:

Open an issue in
CommonMarker

VULNERABLE GEM: COMMONMARKER@0.17.9

Name:
commonmarker

Version:
0.17.9

ID:
GHSA-4qw4-jpp4-8gvp

LINK

Unbounded resource exhaustion in cmark-gfm autolink extension may lead to denial of service

DESCRIPTION:

IMPACT

CommonMarker uses `cmark-gfm` for rendering Github Flavored Markdown. A polynomial time complexity issue in cmark-gfm's autolink extension may lead to unbounded resource exhaustion and subsequent denial of service.

PATCHES

This vulnerability has been patched in the following CommonMarker release:

v0.23.6

WORKAROUNDS

Disable use of the autolink extension.

REFERENCES

https://en.wikipedia.org/wiki/Time_complexity

VULNERABLE GEM: COMMONMARKER@0.17.9

Name:
commonmarker

Version:
0.17.9

ID:
GHSA-fmx4-26r3-wxpf

LINK

Integer overflow in cmark-gfm table parsing extension leads to heap memory corruption

DESCRIPTION:

IMPACT

CommonMarker uses `cmark-gfm` for rendering Github Flavored Markdown. An integer overflow in `cmark-gfm` 's table row parsing may lead to heap memory corruption when parsing tables who's marker rows contain more than UINT16_MAX columns. The impact of this heap corruption ranges from Information Leak to Arbitrary Code Execution.
If affected versions of CommonMarker are used for rendering remote user controlled markdown, this vulnerability may lead to Remote Code Execution (RCE).

PATCHES

This vulnerability has been patched in the following CommonMarker release:

v0.23.4

WORKAROUNDS

The vulnerability exists in the table markdown extensions of `cmark-gfm`. Disabling any use of the table extension will prevent this vulnerability from being triggered.

VULNERABLE GEM: COMMONMARKER@0.17.9

Name:
commonmarker

Version:
0.17.9

ID:
GHSA-7vh7-fw88-wj87

LINK

Several quadratic complexity bugs may lead to denial of service in Commonmarker

DESCRIPTION:

IMPACT

Several quadratic complexity bugs in commonmarker's underlying `cmark-gfm` library may lead to unbounded resource exhaustion and subsequent denial of service.

The following vulnerabilities were addressed:

CVE-2023-37463

For more information, consult the release notes for version `0.29.0.gfm.12`.

MITIGATION

Users are advised to upgrade to commonmarker version `0.23.10`.

VULNERABLE GEM: FFI@1.9.23

Name:
ffi

Version:
1.9.23

ID:
CVE-2018-1000201

LINK

---

ruby-ffi DDL loading issue on Windows OS

DESCRIPTION:

ruby-ffi version 1.9.23 and earlier has a DLL loading issue which can be

hijacked on Windows OS, when a Symbol is used as DLL name instead of a String This vulnerability appears to have been fixed in v1.9.24 and later.

VULNERABLE GEM: JEKYLL@3.7.3

Name:
jekyll

Version:
3.7.3

ID:
CVE-2018-17567

LINK

Jekyll _config.yml privilege escalation

DESCRIPTION:

Jekyll through 3.6.2, 3.7.x through 3.7.3, and 3.8.x through 3.8.3 allows attackers to access arbitrary files by specifying a symlink in the "include" key in the "_config.yml" file.

VULNERABLE GEM: KRAMDOWN@1.16.2

Name:

Version:

kramdown **1.16.2**

ID:
CVE-2020-14001

LINK

Unintended read access in kramdown gem

DESCRIPTION:

The kramdown gem before 2.3.0 for Ruby processes the template option inside Kramdown documents by default, which allows unintended read access (such as template="/etc/passwd") or unintended embedded Ruby code execution (such as a string that begins with template="string://<%= `). NOTE: kramdown is used in Jekyll, GitLab Pages, GitHub Pages, and Thredded Forum.

VULNERABLE GEM: KRAMDOWN@1.16.2

Name:
kramdown

Version:
1.16.2

ID:
CVE-2021-28834

LINK

Remote code execution in Kramdown

DESCRIPTION:

Kramdown before 2.3.1 does not restrict Rouge formatters to the Rouge::Formatters namespace, and thus arbitrary classes can be instantiated.

VULNERABLE GEM: NOKOGIRI@1.8.2

Name:
nokogiri

Version:
1.8.2

ID:
CVE-2021-41098

LINK

Improper Restriction of XML External Entity Reference (XXE) in Nokogiri on JRuby

DESCRIPTION:

SEVERITY

The Nokogiri maintainers have evaluated this as **High Severity** 7.5 (CVSS3.0) for JRuby users. (This security advisory does not apply to CRuby users.)

IMPACT

In Nokogiri v1.12.4 and earlier, **on JRuby only**, the SAX parser resolves external entities by default.
Users of Nokogiri on JRuby who parse untrusted documents using any of these classes are affected:

- **Nokogiri::XML::SAX::Parser**
- **Nokogiri::HTML4::SAX::Parser or its alias**
- **Nokogiri::HTML::SAX::Parser**
- **Nokogiri::XML::SAX::PushParser**
- **Nokogiri::HTML4::SAX::PushParser or its alias**
- **Nokogiri::HTML::SAX::PushParser**

MITIGATION

JRuby users should upgrade to Nokogiri v1.12.5 or later. There are no workarounds available for v1.12.4 or earlier.
CRuby users are not affected.

VULNERABLE GEM: NOKOGIRI@1.8.2

Name:
nokogiri

Version:
1.8.2

ID:
CVE-2021-30560

LINK

Update packaged libxml2 (2.9.12 â† 2.9.13) and libxslt (1.1.34 â† 1.1.35)

DESCRIPTION:

SUMMARY

Nokogiri v1.13.2 upgrades two of its packaged dependencies:

- **vendored libxml2 from v2.9.12 to v2.9.13**
- **vendored libxslt from v1.1.34 to v1.1.35**

Those library versions address the following upstream CVEs:

- **libxslt: CVE-2021-30560 (CVSS 8.8, High severity)**
- **libxml2: CVE-2022-23308 (Unspecified severity, see more information below)**

Those library versions also address numerous other issues including performance improvements, regression fixes, and bug fixes, as well as memory leaks and other use-after-free issues that were not assigned CVEs. Please note that this advisory only applies to the CRuby implementation of Nokogiri < 1.13.2, and only if the packaged libraries are being used. If you've overridden defaults at installation time to use system libraries instead of packaged libraries, you should instead pay attention to your distro's `libxml2` and `libxslt` release announcements.

MITIGATION

Upgrade to Nokogiri >= 1.13.2.

Users who are unable to upgrade Nokogiri may also choose a more complicated mitigation: compile and link an older version Nokogiri against external libraries libxml2 >= 2.9.13 and libxslt >= 1.1.35, which will also address these same CVEs.

IMPACT

libxslt CVE-2021-30560	**CVSS3 score: 8.8 (High)**

Fixed by https://gitlab.gnome.org/GNOME/libxslt/-/commit/50f9c9c
All versions of libxslt prior to v1.1.35 are affected.
Applications using untrusted XSL stylesheets to transform XML are vulnerable to a denial-of-service attack and should be upgraded immediately.
libxml2 CVE-2022-23308 * As of the time this security advisory was published, there is no officially published information available about this CVE's severity. The above NIST link does not yet have a published record, and the libxml2 maintainer has declined to provide a severity score. * Fixed by https://gitlab.gnome.org/GNOME/libxml2/-/commit/652dd12 * Further explanation is at https://mail.gnome.org/archives/xml/2022-February/msg00015.html
The upstream commit and the explanation linked above indicate that an application may be vulnerable to a denial of service, memory disclosure, or code execution if it parses an untrusted document with parse options `DTDVALID` set to true, and `NOENT` set to false.
An analysis of these parse options:

> **While `NOENT` is off by default for Document, DocumentFragment, Reader, and Schema parsing, it is on by default for XSLT (stylesheet) parsing in Nokogiri v1.12.0 and later.**
> **`DTDVALID` is an option that Nokogiri does not set for any operations, and so this CVE applies only to applications setting this option explicitly.**

It seems reasonable to assume that any application explicitly setting the parse option `DTDVALID` when parsing untrusted documents is vulnerable and should be upgraded immediately.

VULNERABLE GEM: NOKOGIRI@1.8.2

Name:
nokogiri

Version:
1.8.2

ID:
CVE-2021-3517

LINK

Nokogiri contains libxml Out-of-bounds Write vulnerability
DESCRIPTION:
There is a flaw in the xml entity encoding functionality of libxml2 in versions

before 2.9.11. An attacker who is able to supply a crafted file to be processed by an application linked with the affected functionality of libxml2 could trigger an out-of-bounds read. The most likely impact of this flaw is to application availability, with some potential impact to confidentiality and integrity if an attacker is able to use memory information to further exploit the application.
Nokogiri prior to version 1.11.4 used a vulnerable version of libxml2. Nokogiri 1.11.4 updated libxml2 to version 2.9.11 to address this and other vulnerabilities in libxml2.

VULNERABLE GEM: NOKOGIRI@1.8.2

Name:
nokogiri

Version:
1.8.2

ID:
CVE-2018-8048

LINK

Revert libxml2 behavior in Nokogiri gem that could cause XSS
DESCRIPTION:
[MRI] Behavior in libxml2 has been reverted which caused CVE-2018-8048 (loofah gem), CVE-2018-3740 (sanitize gem), and CVE-2018-3741 (rails-html-sanitizer gem). The commit in question is here:
https://github.com/GNOME/libxml2/commit/960f0e2
and more information is available about this commit and its impact here:
https://github.com/flavorjones/loofah/issues/144
This release simply reverts the libxml2 commit in question to protect users of Nokogiri's vendored libraries from similar vulnerabilities.
If you're offended by what happened here, I'd kindly ask that you comment on the upstream bug report here:
https://bugzilla.gnome.org/show_bug.cgi?id=769760

VULNERABLE GEM: NOKOGIRI@1.8.2

Name:
nokogiri

Version:
1.8.2

ID:
CVE-2019-5477

LINK

Nokogiri Command Injection Vulnerability via Nokogiri::CSS::Tokenizer#load_file

DESCRIPTION:

A command injection vulnerability in Nokogiri v1.10.3 and earlier allows commands to be executed in a subprocess by Ruby's `Kernel.open` method. Processes are vulnerable only if the undocumented method `Nokogiri::CSS::Tokenizer#load_file` is being passed untrusted user input. This vulnerability appears in code generated by the Rexical gem versions v1.0.6 and earlier. Rexical is used by Nokogiri to generate lexical scanner code for parsing CSS queries. The underlying vulnerability was addressed in Rexical v1.0.7 and Nokogiri upgraded to this version of Rexical in Nokogiri v1.10.4.

Upgrade to Nokogiri v1.10.4, or avoid calling the undocumented method `Nokogiri::CSS::Tokenizer#load_file` with untrusted user input.

VULNERABLE GEM: NOKOGIRI@1.8.2

Name:
nokogiri

Version:
1.8.2

ID:
CVE-2019-5815

LINK

Nokogiri implementation of libxslt vulnerable to heap corruption

DESCRIPTION:

Type confusion in `xsltNumberFormatGetMultipleLevel` prior to libxslt 1.1.33 could allow attackers to potentially exploit heap corruption via crafted XML data.

Nokogiri prior to version 1.10.5 contains a vulnerable version of libxslt. Nokogiri version 1.10.5 upgrades the dependency to libxslt 1.1.34, which contains a patch for this issue.

VULNERABLE GEM: NOKOGIRI@1.8.2

Name:
nokogiri

Version:
1.8.2

ID:
CVE-2020-7595

LINK

---

libxml2 2.9.10 has an infinite loop in a certain end-of-file situation

DESCRIPTION:

Nokogiri has backported the patch for CVE-2020-7595 into its vendored version of libxml2, and released this as v1.10.8
CVE-2020-7595 has not yet been addressed in an upstream libxml2 release, and so Nokogiri versions <= v1.10.7 are vulnerable.

VULNERABLE GEM: NOKOGIRI@1.8.2

Name:
nokogiri

Version:
1.8.2

ID:
CVE-2019-13117

LINK

Nokogiri gem, via libxslt, is affected by multiple vulnerabilities

DESCRIPTION:

Nokogiri v1.10.5 has been released.
This is a security release. It addresses three CVEs in upstream libxml2, for which details are below.
If you're using your distro's system libraries, rather than Nokogiri's vendored libraries, there's no security need to upgrade at this time, though you may want to check with your distro whether they've patched this (Canonical has patched Ubuntu packages). Note that libxslt 1.1.34 addresses these vulnerabilities.
Full details about the security update are available in Github Issue [#1943] https://github.com/sparklemotion/nokogiri/issues/1943.

CVE-2019-13117
https://people.canonical.com/~ubuntu-security/cve/2019/CVE-2019-13117.html
Priority: Low
Description: In numbers.c in libxslt 1.1.33, an xsl:number with certain format strings could lead to a uninitialized read in xsltNumberFormatInsertNumbers. This could allow an attacker to discern whether a byte on the stack contains the characters A, a, I, i, or 0, or any other character.
Patched with commit
https://gitlab.gnome.org/GNOME/libxslt/commit/c5eb6cf3aba0af048596106ec

---

CVE-2019-13118
https://people.canonical.com/~ubuntu-security/cve/2019/CVE-2019-13118.html
Priority: Low
Description: In numbers.c in libxslt 1.1.33, a type holding grouping characters of an xsl:number instruction was too narrow and an invalid character/length combination could be passed to xsltNumberFormatDecimal, leading to a read of uninitialized stack data
Patched with commit
https://gitlab.gnome.org/GNOME/libxslt/commit/6ce8de69330783977dd14f65

CVE-2019-18197
https://people.canonical.com/~ubuntu-security/cve/2019/CVE-2019-18197.html
Priority: Medium
Description: In xsltCopyText in transform.c in libxslt 1.1.33, a pointer variable isn't reset under certain circumstances. If the relevant memory area happened to be freed and reused in a certain way, a bounds check could fail and memory outside a buffer could be written to, or uninitialized data could

be disclosed.
Patched with commit
https://gitlab.gnome.org/GNOME/libxslt/commit/2232473733b7313d67de883

VULNERABLE GEM: NOKOGIRI@1.8.2

Name:
nokogiri

Version:
1.8.2

ID:
GHSA-cgx6-hpwq-fhv5

LINK

Integer Overflow or Wraparound in libxml2 affects Nokogiri

DESCRIPTION:

SUMMARY

Nokogiri v1.13.5 upgrades the packaged version of its dependency libxml2 from v2.9.13 to v2.9.14.
libxml2 v2.9.14 addresses CVE-2022-29824. This version also includes several security-related bug fixes for which CVEs were not created, including a potential double-free, potential memory leaks, and integer-overflow.
Please note that this advisory only applies to the CRuby implementation of Nokogiri `< 1.13.5`, and only if the *packaged* libraries are being used. If you've overridden defaults at installation time to use *system* libraries instead of packaged libraries, you should instead pay attention to your distro's `libxml2` and `libxslt` release announcements.

MITIGATION

Upgrade to Nokogiri `>= 1.13.5`.
Users who are unable to upgrade Nokogiri may also choose a more complicated mitigation: compile and link Nokogiri against external libraries libxml2 `>= 2.9.14` which will also address these same issues.

IMPACT

libxml2 CVE-2022-29824

CVSS3 score:
Unspecified upstream
Nokogiri maintainers evaluate at 8.6 (High) (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:H).
Note that this is different from the CVSS assessed by NVD.

Type: Denial of service, information disclosure
Description: In libxml2 before 2.9.14, several buffer handling functions in buf.c (xmlBuf*) and tree.c (xmlBuffer*) don't check for integer overflows. This can result in out-of-bounds memory writes. Exploitation requires a victim to open a crafted, multi-gigabyte XML file. Other software using libxml2's buffer functions, for example libxslt through 1.1.35, is affected as well.
Fixed: https://gitlab.gnome.org/GNOME/libxml2/-/commit/2554a24

All versions of libml2 prior to v2.9.14 are affected.
Applications parsing or serializing multi-gigabyte documents (in excess of INT_MAX bytes) may be vulnerable to an integer overflow bug in buffer handling that could lead to exposure of confidential data, modification of unrelated data, or a segmentation fault resulting in a denial-of-service.
REFERENCES

libxml2 v2.9.14 release notes **CVE-2022-29824**
CWE-119: Improper Restriction of Operations within the Bounds of a Memory Buffer

VULNERABLE GEM: NOKOGIRI@1.8.2

Name: nokogiri
Version: 1.8.2

ID:

CVE-2020-26247

LINK

Nokogiri::XML::Schema trusts input by default, exposing risk of an XXE vulnerability

DESCRIPTION:

DESCRIPTION

In Nokogiri versions <= 1.11.0.rc3, XML Schemas parsed by `Nokogiri::XML::Schema` are **trusted** by default, allowing external resources to be accessed over the network, potentially enabling XXE or SSRF attacks.
This behavior is counter to the security policy followed by Nokogiri maintainers, which is to treat all input as **untrusted** by default whenever possible.
Please note that this security fix was pushed into a new minor version, 1.11.x, rather than a patch release to the 1.10.x branch, because it is a breaking change for some schemas and the risk was assessed to be "Low Severity".
AFFECTED VERSIONS

Nokogiri `<= 1.10.10` as well as prereleases `1.11.0.rc1`, `1.11.0.rc2`, and `1.11.0.rc3`
MITIGATION

There are no known workarounds for affected versions. Upgrade to Nokogiri `1.11.0.rc4` or later.
If, after upgrading to `1.11.0.rc4` or later, you wish to re-enable network access for resolution of external resources (i.e., return to the previous behavior):

> **Ensure the input is trusted. Do not enable this option for untrusted input.**
> **When invoking the `Nokogiri::XML::Schema` constructor, pass as the second parameter an instance of `Nokogiri::XML::ParseOptions` with the `NONET` flag turned off.**

So if your previous code was:

```
 # in v1.11.0.rc3 and earlier, this call allows resources to be acces
sed over the network
# but in v1.11.0.rc4 and later, this call will disallow network access f
or external resources
schema = Nokogiri::XML::Schema.new(schema)

# in v1.11.0.rc4 and later, the following is equivalent to the code ab
ove
# (the second parameter is optional, and this demonstrates its defa
ult value)
schema = Nokogiri::XML::Schema.new(schema, Nokogiri::XML::Pa
rseOptions::DEFAULT_SCHEMA)
```

Then you can add the second parameter to indicate that the input is trusted by changing it to:

```
 # in v1.11.0.rc3 and earlier, this would raise an ArgumentError
# but in v1.11.0.rc4 and later, this allows resources to be accessed
over the network
schema = Nokogiri::XML::Schema.new(trusted_schema, Nokogiri::
XML::ParseOptions.new.nononet)
```

VULNERABLE GEM: NOKOGIRI@1.8.2

Name:
nokogiri

Version:
1.8.2

ID:
CVE-2021-3518

LINK

Nokogiri Implements libxml2 version vulnerable to use-after-free

DESCRIPTION:

There's a flaw in libxml2 in versions before 2.9.11. An attacker who is able to submit a crafted file to be processed by an application linked with libxml2 could trigger a use-after-free. The greatest impact from this flaw is to

confidentiality, integrity, and availability.

VULNERABLE GEM: NOKOGIRI@1.8.2

Name:
nokogiri

Version:
1.8.2

ID:
CVE-2021-3537

LINK

Nokogiri Implements libxml2 version vulnerable to null pointer dereferencing

DESCRIPTION:

A vulnerability found in libxml2 in versions before 2.9.11 shows that it did not propagate errors while parsing XML mixed content, causing a NULL dereference. If an untrusted XML document was parsed in recovery mode and post-validated, the flaw could be used to crash the application. The highest threat from this vulnerability is to system availability.

VULNERABLE GEM: NOKOGIRI@1.8.2

Name:
nokogiri

Version:
1.8.2

ID:
GHSA-xc9x-jj77-9p9j

LINK

Use-after-free in libxml2 via Nokogiri::XML::Reader

DESCRIPTION:

SUMMARY

Nokogiri upgrades its dependency libxml2 as follows: - v1.15.6 upgrades libxml2 to 2.11.7 from 2.11.6 - v1.16.2 upgrades libxml2 to 2.12.5 from 2.12.4
libxml2 v2.11.7 and v2.12.5 address the following vulnerability: CVE-2024-25062 / https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-25062 - described at https://gitlab.gnome.org/GNOME/libxml2/-/issues/604 - patched by https://gitlab.gnome.org/GNOME/libxml2/-/commit/92721970
Please note that this advisory only applies to the CRuby implementation of Nokogiri, and only if the packaged libraries are being used. If you've overridden defaults at installation time to use system libraries instead of packaged libraries, you should instead pay attention to your distro's libxml2 release announcements.
JRuby users are not affected.
SEVERITY

The Nokogiri maintainers have evaluated this as **Moderate**.
IMPACT

From the CVE description, this issue applies to the `xmlTextReader` module (which underlies `Nokogiri::XML::Reader`):
When using the XML Reader interface with DTD validation and XInclude expansion enabled, processing crafted XML documents can lead to an xmlValidatePopElement use-after-free.
MITIGATION

Upgrade to Nokogiri `~> 1.15.6` or `>= 1.16.2`.
Users who are unable to upgrade Nokogiri may also choose a more complicated mitigation: compile and link Nokogiri against patched external libxml2 libraries which will also address these same issues.

VULNERABLE GEM: NOKOGIRI@1.8.2

Name:
nokogiri

Version:
1.8.2

ID:
CVE-2022-24836

LINK

Inefficient Regular Expression Complexity in Nokogiri

DESCRIPTION:

SUMMARY

Nokogiri `< v1.13.4` contains an inefficient regular expression that is susceptible to excessive backtracking when attempting to detect encoding in HTML documents.

MITIGATION

Upgrade to Nokogiri `>= 1.13.4`.

VULNERABLE GEM: NOKOGIRI@1.8.2

Name:
nokogiri

Version:
1.8.2

ID:
CVE-2018-25032

LINK

Out-of-bounds Write in zlib affects Nokogiri

DESCRIPTION:

SUMMARY

Nokogiri v1.13.4 updates the vendored zlib from 1.2.11 to 1.2.12, which addresses CVE-2018-25032. That CVE is scored as CVSS 7.4 "High" on the NVD record as of 2022-04-05.

Please note that this advisory only applies to the CRuby implementation of Nokogiri `< 1.13.4`, and only if the packaged version of `zlib` is being used. Please see this document for a complete description of which platform

gems vendor `zlib`. If you've overridden defaults at installation time to use system libraries instead of packaged libraries, you should instead pay attention to your distro's `zlib` release announcements.

MITIGATION

Upgrade to Nokogiri `>= v1.13.4`.

IMPACT

CVE-2018-25032 IN ZLIB

Type: CWE-787 Out of bounds write

Severity: High

Description: zlib before 1.2.12 allows memory corruption when deflating (i.e., when compressing) if the input has many distant matches.

VULNERABLE GEM: NOKOGIRI@1.8.2

Name:
nokogiri

Version:
1.8.2

ID:
CVE-2019-11068

LINK

Nokogiri gem, via libxslt, is affected by improper access control vulnerability

DESCRIPTION:

Nokogiri v1.10.3 has been released.
This is a security release. It addresses a CVE in upstream libxslt rated as "Priority: medium" by Canonical, and "NVD Severity: high" by Debian. More details are available below.
If you're using your distro's system libraries, rather than Nokogiri's vendored libraries, there's no security need to upgrade at this time, though you may want to check with your distro whether they've patched this (Canonical has patched Ubuntu packages). Note that this patch is not yet (as of 2019-04-22) in an upstream release of libxslt.
Full details about the security update are available in Github Issue [#1892] https://github.com/sparklemotion/nokogiri/issues/1892.

CVE-2019-11068
Permalinks are: - Canonical: https://people.canonical.com/~ubuntu-security/cve/CVE-2019-11068 - Debian: https://security-tracker.debian.org/tracker/CVE-2019-11068
Description:
libxslt through 1.1.33 allows bypass of a protection mechanism because callers of xsltCheckRead and xsltCheckWrite permit access even upon receiving a -1 error code. xsltCheckRead can return -1 for a crafted URL that is not actually invalid and is subsequently loaded.
Canonical rates this as "Priority: Medium".
Debian rates this as "NVD Severity: High (attack range: remote)".

VULNERABLE GEM: NOKOGIRI@1.8.2

Name:
nokogiri

Version:
1.8.2

ID:
GHSA-pxvg-2qj5-37jq

LINK

Update packaged libxml2 to v2.10.4 to resolve multiple CVEs
DESCRIPTION:
SUMMARY

Nokogiri v1.14.3 upgrades the packaged version of its dependency libxml2 to v2.10.4 from v2.10.3.
libxml2 v2.10.4 addresses the following known vulnerabilities:

- **CVE-2023-29469: Hashing of empty dict strings isn't deterministic**
- **CVE-2023-28484: Fix null deref in xmlSchemaFixupComplexType**
- **Schemas: Fix null-pointer-deref in xmlSchemaCheckCOSSTDerivedOK**

Please note that this advisory only applies to the CRuby implementation of Nokogiri `< 1.14.3`, and only if the *packaged* libraries are being used. If you've overridden defaults at installation time to use *system* libraries instead of packaged libraries, you should instead pay attention to your distro's

`libxml2` release announcements.
MITIGATION

Upgrade to Nokogiri `>= 1.14.3`.
Users who are unable to upgrade Nokogiri may also choose a more complicated mitigation: compile and link Nokogiri against external libraries libxml2 `>= 2.10.4` which will also address these same issues.
IMPACT

No public information has yet been published about the security-related issues other than the upstream commits. Examination of those changesets indicate that the more serious issues relate to libxml2 dereferencing NULL pointers and potentially segfaulting while parsing untrusted inputs.
The commits can be examined at:

- **[CVE-2023-29469] Hashing of empty dict strings isn't deterministic (09a2dd45)**
- **[CVE-2023-28484] Fix null deref in xmlSchemaFixupComplexType (647e072e)**
- **schemas: Fix null-pointer-deref in xmlSchemaCheckCOSSTDerivedOK (4c6922f7)**

VULNERABLE GEM: NOKOGIRI@1.8.2

Name:
nokogiri

Version:
1.8.2

ID:
CVE-2022-29181

LINK

Improper Handling of Unexpected Data Type in Nokogiri
DESCRIPTION:
SUMMARY

Nokogiri `< v1.13.6` does not type-check all inputs into the XML and HTML4 SAX parsers. For CRuby users, this may allow specially crafted untrusted inputs to cause illegal memory access errors (segfault) or reads from unrelated memory.

SEVERITY

The Nokogiri maintainers have evaluated this as **High 8.2** (CVSS3.1).
MITIGATION

CRuby users should upgrade to Nokogiri `>= 1.13.6`.
JRuby users are not affected.
WORKAROUNDS

To avoid this vulnerability in affected applications, ensure the untrusted input is a `String` by calling `#to_s` or equivalent.

VULNERABLE GEM: NOKOGIRI@1.8.2

Name:
nokogiri

Version:
1.8.2

ID:
GHSA-2qc6-mcvw-92cw

LINK

Update bundled libxml2 to v2.10.3 to resolve multiple CVEs
DESCRIPTION:
SUMMARY

Nokogiri v1.13.9 upgrades the packaged version of its dependency libxml2 to v2.10.3 from v2.9.14.
libxml2 v2.10.3 addresses the following known vulnerabilities:
CVE-2022-2309 **CVE-2022-40304** **CVE-2022-40303**

Please note that this advisory only applies to the CRuby implementation of Nokogiri `< 1.13.9`, and only if the *packaged* libraries are being used. If you've overridden defaults at installation time to use *system* libraries instead of packaged libraries, you should instead pay attention to your distro's `libxml2` release announcements.
MITIGATION

Upgrade to Nokogiri `>= 1.13.9`.
Users who are unable to upgrade Nokogiri may also choose a more

complicated mitigation: compile and link Nokogiri against external libraries libxml2 `>= 2.10.3` which will also address these same issues.
IMPACT

libxml2 CVE-2022-2309

CVSS3 score: Under evaluation
Type: Denial of service
Description: NULL Pointer Dereference allows attackers to cause a denial of service (or application crash). This only applies when lxml is used together with libxml2 2.9.10 through 2.9.14. libxml2 2.9.9 and earlier are not affected. It allows triggering crashes through forged input data, given a vulnerable code sequence in the application. The vulnerability is caused by the iterwalk function (also used by the canonicalize function). Such code shouldn't be in wide-spread use, given that parsing + iterwalk would usually be replaced with the more efficient iterparse function. However, an XML converter that serialises to C14N would also be vulnerable, for example, and there are legitimate use cases for this code sequence. If untrusted input is received (also remotely) and processed via iterwalk function, a crash can be triggered.

Nokogiri maintainers investigated at #2620 and determined this CVE does not affect Nokogiri users.

libxml2 CVE-2022-40304

CVSS3 score: Unspecified upstream
Type: Data corruption, denial of service
Description: When an entity reference cycle is detected, the entity content is cleared by setting its first byte to zero. But the entity content might be allocated from a dict. In this case, the dict entry becomes corrupted leading to all kinds of logic errors, including memory errors like double-frees.

See https://gitlab.gnome.org/GNOME/libxml2/-/commit/644a89e080bced793295f61f18aac8cfad6bece2

libxml2 CVE-2022-40303

CVSS3 score: Unspecified upstream
Type: Integer overflow
Description: Integer overflows with XML*PARSE*HUGE

See https://gitlab.gnome.org/GNOME/libxml2/-/commit/c846986356fc149915a74972bf198abc266bc2c0

VULNERABLE GEM: NOKOGIRI@1.8.2

Name:
nokogiri

Version:
1.8.2

ID:
CVE-2022-23437

LINK

XML Injection in Xerces Java affects Nokogiri

DESCRIPTION:

SUMMARY

Nokogiri v1.13.4 updates the vendored `xerces:xercesImpl` from 2.12.0 to 2.12.2, which addresses CVE-2022-23437. That CVE is scored as CVSS 6.5 "Medium" on the NVD record.

Please note that this advisory only applies to the **JRuby** implementation of Nokogiri `< 1.13.4`.

MITIGATION

Upgrade to Nokogiri `>= v1.13.4`.

IMPACT

CVE-2022-23437 IN XERCES-J

Severity:
Medium
Type: CWE-91 XML Injection (aka Blind XPath Injection)
Description: There's a vulnerability within the Apache Xerces Java (XercesJ) XML parser when handling specially crafted XML document payloads. This causes, the XercesJ XML parser to wait in an infinite loop, which may sometimes consume system resources for prolonged duration. This vulnerability is present within XercesJ version 2.12.1 and the previous versions.
See also: https://github.com/advisories/GHSA-h65f-jvqw-m9fj

VULNERABLE GEM: NOKOGIRI@1.8.2

Name:
nokogiri

Version:
1.8.2

ID:
CVE-2022-24839

LINK

Denial of Service (DoS) in Nokogiri on JRuby

DESCRIPTION:

SUMMARY

Nokogiri `v1.13.4` updates the vendored `org.cyberneko.html` library to `1.9.22.noko2` which addresses CVE-2022-24839. That CVE is rated 7.5 (High Severity).

See GHSA-9849-p7jc-9rmv for more information.

Please note that this advisory only applies to the **JRuby** implementation of Nokogiri `< 1.13.4`.

MITIGATION

Upgrade to Nokogiri `>= 1.13.4`.

IMPACT

CVE-2022-24839 IN NEKOHTML

Severity: High
7.5
Type: CWE-400 Uncontrolled Resource Consumption
Description: The fork of `org.cyberneko.html` used by Nokogiri (Rubygem) raises a `java.lang.OutOfMemoryError` exception when parsing ill-formed HTML markup.
See also: GHSA-9849-p7jc-9rmv

VULNERABLE GEM: NOKOGIRI@1.8.2

Name:
nokogiri

Version:
1.8.2

ID:
CVE-2019-18197

LINK

Nokogiri affected by libxslt Use of Uninitialized Resource/ Use After Free vulnerability

DESCRIPTION:

In xsltCopyText in transform.c in libxslt 1.1.33, a pointer variable isn't reset under certain circumstances. If the relevant memory area happened to be freed and reused in a certain way, a bounds check could fail and memory outside a buffer could be written to, or uninitialized data could be disclosed. Nokogiri prior to version 1.10.5 contains a vulnerable version of libxslt. Nokogiri version 1.10.5 upgrades the dependency to libxslt 1.1.34, which contains a patch for this issue.

VULNERABLE GEM: NOKOGIRI@1.8.2

Name:
nokogiri

Version:
1.8.2

ID:
GHSA-7rrm-v45f-jp64

LINK

Update packaged dependency libxml2 from 2.9.10 to 2.9.12

DESCRIPTION:

SUMMARY

Nokogiri v1.11.4 updates the vendored libxml2 from v2.9.10 to v2.9.12 which addresses:

CVE-2019-20388 (Medium severity)
CVE-2020-24977 (Medium severity)
CVE-2021-3517 (Medium severity)
CVE-2021-3518 (Medium severity)
CVE-2021-3537 (Low severity)
CVE-2021-3541 (Low severity)

Note that two additional CVEs were addressed upstream but are not relevant to this release. CVE-2021-3516 via `xmllint` is not present in Nokogiri, and CVE-2020-7595 has been patched in Nokogiri since v1.10.8 (see #1992).
Please note that this advisory only applies to the CRuby implementation of Nokogiri `< 1.11.4`, and only if the packaged version of libxml2 is being used. If you've overridden defaults at installation time to use system libraries instead of packaged libraries, you should instead pay attention to your distro's `libxml2` release announcements.
MITIGATION

Upgrade to Nokogiri `>= 1.11.4`.
IMPACT

I've done a brief analysis of the published CVEs that are addressed in this upstream release. The libxml2 maintainers have not released a canonical set of CVEs, and so this list is pieced together from secondary sources and may be incomplete.
All information below is sourced from security.archlinux.org, which appears to have the most up-to-date information as of this analysis.
CVE-2019-20388

Severity: Medium **Type: Denial of service**
Description: A memory leak was found in the xmlSchemaValidateStream function of libxml2. Applications that use this library may be vulnerable to memory not being freed leading to a denial of service.
Fixed: https://gitlab.gnome.org/GNOME/libxml2/commit/7ffcd44d7e6

Verified that the fix commit first appears in v2.9.11. It seems possible that this issue would be present in programs using Nokogiri < v1.11.4.
CVE-2020-7595

Severity: Medium **Type: Denial of service**
Description: xmlStringLenDecodeEntities in parser.c in libxml2 2.9.10 has an infinite loop in a certain end-of-file situation.

Fixed:
https://gitlab.gnome.org/GNOME/libxml2/commit/0e1a49c890

This has been patched in Nokogiri since v1.10.8 (see #1992).

CVE-2020-24977

Severity:
Medium **Type: Information disclosure**
Description: GNOME project libxml2 <= 2.9.10 has a global buffer over-read vulnerability in xmlEncodeEntitiesInternal at libxml2/entities.c.
Fixed:
https://gitlab.gnome.org/GNOME/libxml2/commit/50f06b3efb6

Verified that the fix commit first appears in v2.9.11. It seems possible that this issue would be present in programs using Nokogiri < v1.11.4.

CVE-2021-3516

Severity:
Medium
Type: Arbitrary code execution (no remote vector)
Description: A use-after-free security issue was found libxml2 before version 2.9.11 when "xmllint --html --push" is used to process crafted files.
Issue: https://gitlab.gnome.org/GNOME/libxml2/-/issues/230
Fixed: https://gitlab.gnome.org/GNOME/libxml2/-/commit/1358d157d0bd83be1dfe356a69213df9fac0b539

Verified that the fix commit first appears in v2.9.11. This vector does not exist within Nokogiri, which does not ship `xmllint`.

CVE-2021-3517

Severity: **Type: Arbitrary code execution**
Medium
Description: A heap-based buffer overflow was found in libxml2 before version 2.9.11 when processing truncated UTF-8 input.
Issue: https://gitlab.gnome.org/GNOME/libxml2/-/issues/235
Fixed: https://gitlab.gnome.org/GNOME/libxml2/-/commit/bf22713507fe1fc3a2c4b525cf0a88c2dc87a3a2

Verified that the fix commit first appears in v2.9.11. It seems possible that this issue would be present in programs using Nokogiri < v1.11.4.

CVE-2021-3518

Severity: Medium **Type: Arbitrary code execution**

Description: A use-after-free security issue was found in libxml2 before version 2.9.11 in xmlXIncludeDoProcess() in xinclude.c when processing crafted files.

Issue: https://gitlab.gnome.org/GNOME/libxml2/-/issues/237

Fixed: https://gitlab.gnome.org/GNOME/libxml2/-/commit/1098c30a040e72a4654968547f415be4e4c40fe7

Verified that the fix commit first appears in v2.9.11. It seems possible that this issue would be present in programs using Nokogiri < v1.11.4.

CVE-2021-3537

Severity: Low **Type: Denial of service**

Description: It was found that libxml2 before version 2.9.11 did not propagate errors while parsing XML mixed content, causing a NULL dereference. If an untrusted XML document was parsed in recovery mode and post-validated, the flaw could be used to crash the application.

Issue: https://gitlab.gnome.org/GNOME/libxml2/-/issues/243

Fixed: https://gitlab.gnome.org/GNOME/libxml2/-/commit/babe75030c7f64a37826bb3342317134568bef61

Verified that the fix commit first appears in v2.9.11. It seems possible that this issue would be present in programs using Nokogiri < v1.11.4.

CVE-2021-3541

Severity: Low **Type: Denial of service**

Description: A security issue was found in libxml2 before version 2.9.11. Exponential entity expansion attack its possible bypassing all existing protection mechanisms and leading to denial of service.

Fixed: https://gitlab.gnome.org/GNOME/libxml2/-/commit/8598060bacada41a0eb09d95c97744ff4e428f8e

Verified that the fix commit first appears in v2.9.11. It seems possible that this issue would be present in programs using Nokogiri < v1.11.4, however Nokogiri's default parse options prevent the attack from succeeding (it is necessary to opt into `DTDLOAD` which is off by default).
For more details supporting this analysis of this CVE, please visit #2233.

VULNERABLE GEM: NOKOGIRI@1.8.2

Name:
nokogiri

Version:
1.8.2

ID:
CVE-2019-13118

LINK

libxslt Type Confusion vulnerability that affects Nokogiri

DESCRIPTION:

In `numbers.c` in libxslt 1.1.33, a type holding grouping characters of an `xsl:number` instruction was too narrow and an invalid character/length combination could be passed to `xsltNumberFormatDecimal`, leading to a read of uninitialized stack data.
Nokogiri prior to version 1.10.5 used a vulnerable version of libxslt. Nokogiri 1.10.5 updated libxslt to version 1.1.34 to address this and other vulnerabilities in libxslt.

VULNERABLE GEM: NOKOGIRI@1.8.2

Name:
nokogiri

Version:
1.8.2

ID:
CVE-2018-14404

LINK

Nokogiri gem, via libxml2, is affected by multiple vulnerabilities

DESCRIPTION:

Nokogiri 1.8.5 has been released.
This is a security and bugfix release. It addresses two CVEs in upstream

libxml2 rated as "medium" by Red Hat, for which details are below.
If you're using your distro's system libraries, rather than Nokogiri's vendored libraries, there's no security need to upgrade at this time, though you may want to check with your distro whether they've patched this (Canonical has patched Ubuntu packages). Note that these patches are not yet (as of 2018-10-04) in an upstream release of libxml2.
Full details about the security update are available in Github Issue #1785.

[MRI] Pulled in upstream patches from libxml2 that address CVE-2018-14404 and CVE-2018-14567. Full details are available in #1785. Note that these patches are not yet (as of 2018-10-04) in an upstream release of libxml2.

CVE-2018-14404
Permalink:
https://people.canonical.com/~ubuntu-security/cve/2018/CVE-2018-14404.html
Description:
A NULL pointer dereference vulnerability exists in the xpath.c:xmlXPathCompOpEval() function of libxml2 through 2.9.8 when parsing an invalid XPath expression in the XPATH*OP*AND or XPATH*OP*OR case. Applications processing untrusted XSL format inputs with the use of the libxml2 library may be vulnerable to a denial of service attack due to a crash of the application
Canonical rates this vulnerability as "Priority: Medium"

CVE-2018-14567
Permalink:
https://people.canonical.com/~ubuntu-security/cve/2018/CVE-2018-14567.html
Description:
infinite loop in LZMA decompression
Canonical rates this vulnerability as "Priority: Medium"

VULNERABLE GEM: RUBYZIP@1.2.1

Name:
rubyzip

Version:
1.2.1

ID:

CVE-2018-1000544

LINK

Directory Traversal in rubyzip

DESCRIPTION:

rubyzip version 1.2.1 and earlier contains a Directory Traversal vulnerability in Zip::File component that can result in write arbitrary files to the filesystem. If a site allows uploading of .zip files, an attacker can upload a malicious file which contains symlinks or files with absolute pathnames "../" to write arbitrary files to the filesystem.

VULNERABLE GEM: RUBYZIP@1.2.1

Name:
rubyzip

Version:
1.2.1

ID:
CVE-2019-16892

LINK

Denial of Service in rubyzip ("zip bombs")

DESCRIPTION:

In Rubyzip before 1.3.0, a crafted ZIP file can bypass application checks on ZIP entry sizes because data about the uncompressed size can be spoofed. This allows attackers to cause a denial of service (disk consumption).

VULNERABLE GEM: TZINFO@1.2.5

Name:
tzinfo

Version:
1.2.5

ID:
CVE-2022-31163

LINK

TZInfo relative path traversal vulnerability allows loading of arbitrary files

DESCRIPTION:

Impact

AFFECTED VERSIONS

0.3.60 and earlier.

1.0.0 to 1.2.9 when used with the Ruby data source (tzinfo-data).

VULNERABILITY

With the Ruby data source (the tzinfo-data gem for tzinfo version 1.0.0 and later and built-in to earlier versions), time zones are defined in Ruby files. There is one file per time zone. Time zone files are loaded with `require` on demand. In the affected versions, `TZInfo::Timezone.get` fails to validate time zone identifiers correctly, allowing a new line character within the identifier. With Ruby version 1.9.3 and later, `TZInfo::Timezone.get` can be made to load unintended files with `require`, executing them within the Ruby process.

For example, with version 1.2.9, you can run the following to load a file with path `/tmp/payload.rb`:

```
TZInfo::Timezone.get(\"foo\
/../../../../../../../../../../../../../../../../tmp/payload\")
```

The exact number of parent directory traversals needed will vary depending on the location of the tzinfo-data gem.

TZInfo versions 1.2.6 to 1.2.9 can be made to load files from outside of the Ruby load path. Versions up to and including 1.2.5 can only be made to load files from directories within the load path.

This could be exploited in, for example, a Ruby on Rails application using tzinfo version 1.2.9, that allows file uploads and has a time zone selector that accepts arbitrary time zone identifiers. The CVSS score and severity have

been set on this basis.
Versions 2.0.0 and later are not vulnerable.

Patches

Versions 0.3.61 and 1.2.10 include fixes to correctly validate time zone identifiers.
Note that version 0.3.61 can still load arbitrary files from the Ruby load path if their name follows the rules for a valid time zone identifier and the file has a prefix of `tzinfo/definition` within a directory in the load path. For example if `/tmp/upload` was in the load path, then `TZInfo::Timezone.get('foo')` could load a file with path `/tmp/upload/tzinfo/definition/foo.rb`. Applications should ensure that untrusted files are not placed in a directory on the load path.

Workarounds

As a workaround, the time zone identifier can be validated before passing to `TZInfo::Timezone.get` by ensuring it matches the regular expression `\\A[A-Za-z0-9+\\-_]+(?:\\/[A-Za-z0-9+\\-_]+)*\\z`.