FastRuby
Audit Tool

108 Vulnerabilities found on your file

Advisories

108

VULNERABLE GEM: ACTIONMAILER@3.2.22.5

Name:
actionmailer

Version:
3.2.22.5

ID:
CVE-2024-47889

LINK

Possible ReDoS vulnerability in block_format in Action Mailer

DESCRIPTION:

There is a possible ReDoS vulnerability in the block_format helper in Action Mailer. This vulnerability has been assigned the CVE identifier CVE-2024-47889.

IMPACT

Carefully crafted text can cause the block_format helper to take an unexpected amount of time, possibly resulting in a DoS vulnerability. All users running an affected release should either upgrade or apply the relevant patch immediately.

Ruby 3.2 has mitigations for this problem, so Rails applications using Ruby 3.2 or newer are unaffected. Rails 8.0.0.beta1 requires Ruby 3.2 or greater so is unaffected.

RELEASES

The fixed releases are available at the normal locations.

WORKAROUNDS

Users can avoid calling the `block_format` helper or upgrade to Ruby 3.2.

CREDITS

Thanks to oooooo_q for the report!

VULNERABLE GEM: ACTIONPACK@3.2.22.5

Name:
actionpack

Version:
3.2.22.5

ID:
CVE-2023-28362

LINK

Possible XSS via User Supplied Values to redirect_to

DESCRIPTION:

The redirect_to method in Rails allows provided values to contain characters which are not legal in an HTTP header value. This results in the potential for downstream services which enforce RFC compliance on HTTP response headers to remove the assigned Location header. This vulnerability has been assigned the CVE identifier CVE-2023-28362.
Versions Affected: All. Not affected: None Fixed Versions: 7.0.5.1, 6.1.7.4

Impact

This introduces the potential for a Cross-site-scripting (XSS) payload to be delivered on the now static redirection page. Note that this both requires user interaction and for a Rails app to be configured to allow redirects to external hosts (defaults to false in Rails >= 7.0.x).

Releases

The FIXED releases are available at the normal locations.

Workarounds

Avoid providing user supplied URLs with arbitrary schemes to the redirect_to method.

VULNERABLE GEM: ACTIONPACK@3.2.22.5

VULNERABLE GEM: ACTIONPACK@3.2.22.5

Name:
actionpack

Version:
3.2.22.5

ID:
CVE-2023-22792

LINK

ReDoS based DoS vulnerability in Action Dispatch

DESCRIPTION:

There is a possible regular expression based DoS vulnerability in Action Dispatch. This vulnerability has been assigned the CVE identifier CVE-2023-22792.

Versions Affected: >= 3.0.0 Not affected: < 3.0.0 Fixed Versions: 6.1.7.1, 7.0.4.1

Impact

Specially crafted cookies, in combination with a specially crafted X*FORWARDED*HOST header 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 All users running an affected release should either upgrade or use one of the workarounds immediately.

Workarounds

We recommend that all users upgrade to one of the FIXED versions. In the meantime, users can mitigate this vulnerability by using a load balancer or other device to filter out malicious X*FORWARDED*HOST headers before they reach the application.

VULNERABLE GEM: ACTIONPACK@3.2.22.5

Name:
actionpack

Version:
3.2.22.5

ID:
CVE-2024-41128

LINK

Possible ReDoS vulnerability in query parameter filtering in Action

Possible ReDoS vulnerability in query parameter filtering in Action Dispatch

DESCRIPTION:

There is a possible ReDoS vulnerability in the query parameter filtering routines of Action Dispatch. This vulnerability has been assigned the CVE identifier CVE-2024-41128.

IMPACT

Carefully crafted query parameters can cause query parameter filtering to take an unexpected amount of time, possibly resulting in a DoS vulnerability. All users running an affected release should either upgrade or apply the relevant patch immediately.

Ruby 3.2 has mitigations for this problem, so Rails applications using Ruby 3.2 or newer are unaffected. Rails 8.0.0.beta1 depends on Ruby 3.2 or greater so is unaffected.

RELEASES

The fixed releases are available at the normal locations.

WORKAROUNDS

Users on Ruby 3.2 are unaffected by this issue.

CREDITS

Thanks to scyoon for the report and patches!

VULNERABLE GEM: ACTIONPACK@3.2.22.5

Name:
actionpack

Version:
3.2.22.5

ID:
CVE-2023-22795

LINK

ReDoS based DoS vulnerability in Action Dispatch

DESCRIPTION:

There is a possible regular expression based DoS vulnerability in Action Dispatch related to the If-None-Match header. This vulnerability has been assigned the CVE identifier CVE-2023-22795.

Versions Affected: All Not affected: None Fixed Versions: 6.1.7.1, 7.0.4.1

Impact

A specially crafted HTTP If-None-Match header can cause the regular expression engine to enter a state of catastrophic backtracking, when on a version of Ruby below 3.2.0. This can cause the process to use large amounts of CPU and memory, leading to a possible DoS vulnerability All

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

Workarounds

We recommend that all users upgrade to one of the FIXED versions. In the meantime, users can mitigate this vulnerability by using a load balancer or other device to filter out malicious If-None-Match headers before they reach the application.
Users on Ruby 3.2.0 or greater are not affected by this vulnerability.

VULNERABLE GEM: ACTIONPACK@3.2.22.5

Name:
actionpack

Version:
3.2.22.5

ID:
CVE-2020-8166

LINK

Ability to forge per-form CSRF tokens given a global CSRF token

DESCRIPTION:

It is possible to possible to, given a global CSRF token such as the one present in the authenticity_token meta tag, forge a per-form CSRF token for any action for that session.
Versions Affected: rails < 5.2.5, rails < 6.0.4 Not affected: Applications without existing HTML injection vulnerabilities. Fixed Versions: rails >= 5.2.4.3, rails >= 6.0.3.1

IMPACT

Given the ability to extract the global CSRF token, an attacker would be able to construct a per-form CSRF token for that session.

WORKAROUNDS

This is a low-severity security issue. As such, no workaround is necessarily until such time as the application can be upgraded.

VULNERABLE GEM: ACTIONPACK@3.2.22.5

Name:
actionpack

Version:
3.2.22.5

ID:
CVE-2021-22885

LINK

Possible Information Disclosure / Unintended Method Execution in Action Pack

DESCRIPTION:

There is a possible information disclosure / unintended method execution vulnerability in Action Pack which has been assigned the CVE identifier CVE-2021-22885.
Versions Affected: >= 2.0.0. Not affected: < 2.0.0. Fixed Versions: 6.1.3.2, 6.0.3.7, 5.2.4.6, 5.2.6

IMPACT

There is a possible information disclosure / unintended method execution vulnerability in Action Pack when using the `redirect_to` or `polymorphic_url` helper with untrusted user input.
Vulnerable code will look like this:

```
redirect_to(params[:some_param])
```

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

WORKAROUNDS

To work around this problem, it is recommended to use an allow list for valid parameters passed from the user. For example:

```
private def check(param)
case param
when "valid"
  param
else
  "/"
end
end

def index
redirect_to(check(params[:some_param]))
end
```

Or force the user input to be cast to a string like this:

```
 def index
 redirect_to(params[:some_param].to_s)
end
```

VULNERABLE GEM: ACTIVERECORD@3.2.22.5

Name:
activerecord

Version:
3.2.22.5

ID:
CVE-2022-32224

LINK

Possible RCE escalation bug with Serialized Columns in Active Record

DESCRIPTION:

There is a possible escalation to RCE when using YAML serialized columns in Active Record. This vulnerability has been assigned the CVE identifier CVE-2022-32224.
Versions Affected: All. Not affected: None Fixed Versions: 7.0.3.1, 6.1.6.1, 6.0.5.1, 5.2.8.1

IMPACT

When serialized columns that use YAML (the default) are deserialized, Rails uses `YAML.unsafe_load` to convert the YAML data in to Ruby objects. If an attacker can manipulate data in the database (via means like SQL injection), then it may be possible for the attacker to escalate to an RCE.
Impacted Active Record models will look something like this:

```
 class User < ApplicationRecord
 serialize :options      # Vulnerable: Uses YAML for serialization
 serialize :values, Array # Vulnerable: Uses YAML for serialization
 serialize :values, JSON  # Not vulnerable
end
```

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

RELEASES

The FIXED releases are available at the normal locations.

The released versions change the default YAML deserializer to use `YAML.safe_load`, which prevents deserialization of possibly dangerous objects. This may introduce backwards compatibility issues with existing data.
In order to cope with that situation, the released version also contains two new Active Record configuration options. The configuration options are as follows:

`config.active_record.use_yaml_unsafe_load`

When set to true, this configuration option tells Rails to use the old "unsafe" YAML loading strategy, maintaining the existing behavior but leaving the possible escalation vulnerability in place. Setting this option to true is *not* recommended, but can aid in upgrading.

`config.active_record.yaml_column_permitted_classes`

The "safe YAML" loading method does not allow all classes to be deserialized by default. This option allows you to specify classes deemed "safe" in your application. For example, if your application uses Symbol and Time in serialized data, you can add Symbol and Time to the allowed list as follows:

```
config.active_record.yaml_column_permitted_classes = [Symbol, Date, Time]
```

WORKAROUNDS
There are no feasible workarounds for this issue, but other coders (such as JSON) are not impacted.

VULNERABLE GEM: ACTIVERECORD@3.2.22.5

Name:
activerecord

Version:
3.2.22.5

ID:
CVE-2022-44566

LINK

Denial of Service Vulnerability in ActiveRecordâ€™s PostgreSQL adapter
DESCRIPTION:

There is a potential denial of service vulnerability present in ActiveRecordâ€™s PostgreSQL adapter.
This has been assigned the CVE identifier CVE-2022-44566.
Versions Affected: All. Not affected: None. Fixed Versions: 6.1.7.1, 7.0.4.1

Impact

In ActiveRecord <7.0.4.1 and <6.1.7.1, when a value outside the range for a 64bit signed integer is provided to the PostgreSQL connection adapter, it will treat the target column type as numeric. Comparing integer values against numeric values can result in a slow sequential scan resulting in potential Denial of Service.

Workarounds

Ensure that user supplied input which is provided to ActiveRecord clauses do not contain integers wider than a signed 64bit representation or floats.

VULNERABLE GEM: ACTIVERESOURCE@3.2.22.5

Name:
activeresource

Version:
3.2.22.5

ID:
CVE-2020-8151

LINK

---

activeresource Gem for Ruby lib/active_resource/base.rb element_path Lack of Encoding

DESCRIPTION:

activeresource contains a lack of encoding flaw in the element *path function of lib/active*resource/base.rb.
There is an issue with the way Active Resource encodes data before querying the back end server. This encoding mechanism can allow specially crafted requests to possibly access data that may not be expected.
Impacted code will look something like this:

```
 require 'activeresource'

class Test < ActiveResource::Base
 self.site = 'http://127.0.0.1:3000'
end

Test.exists?(untrusted_user_input)
```

Where untrusted user input is passed to an Active Resource model. Specially crafted untrusted input can cause Active Resource to access data in an unexpected way and possibly leak information.

WORKAROUNDS

For those that can't upgrade, the following monkey patch can be applied:

```
 module ActiveResource
 class Base
  class << self
   def element_path(id, prefix_options = {}, query_options = nil)
     check_prefix_options(prefix_options)

     prefix_options, query_options = split_options(prefix_options) if query_options.nil?
     "#{prefix(prefix_options)}#{collection_name}/#{URI.encode_www_form_component(id.to_s)}#{format_extension}#{query_string(query_options)}"
   end
  end
 end
end
```

VULNERABLE GEM: ACTIVESUPPORT@3.2.22.5

Name:
activesupport

Version:
3.2.22.5

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: ACTIVESUPPORT@3.2.22.5

Name:	Version:
activesupport	3.2.22.5

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: 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@3.2.22.5

Name:
activesupport

Version:
3.2.22.5

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: ADDRESSABLE@2.3.6

Name:
addressable

Version:
2.3.6

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: BETTER_ERRORS@1.0.1

Name:
better_errors

Version:
1.0.1

ID:
CVE-2021-39197

LINK

Older releases of better_errors open to Cross-Site Request Forgery attack

DESCRIPTION:

IMPACT

better*errors prior to 2.8.0 did not implement CSRF protection for its internal requests. It also did not enforce the correct "Content-Type" header for these requests, which allowed a cross-origin "simple request" to be made without CORS protection. These together left an application with better*errors enabled open to cross-origin attacks.
*As a developer tool, better*errors documentation strongly recommends addition only to the `development` bundle group, so this vulnerability should only affect development environments. Please ensure that your project limits better*errors to the `development` group (or the non-Rails equivalent).*

PATCHES

Starting with release 2.8.x, CSRF protection is enforced. It is recommended that you upgrade to the latest release, or minimally to "~> 2.8.3".

WORKAROUNDS

There are no known workarounds to mitigate the risk of using older releases of better_errors.

REFERENCES

> **Chris Moberly provided an example attack that uses a now-patched vulnerability of webpack-dev-server in conjunction with Better Errors**

FOR MORE INFORMATION

If you have any questions or comments about this advisory, please - Add to

the discussion in better_errors - Open an issue in better_errors

VULNERABLE GEM: COCAINE@0.4.2

Name:
cocaine

Version:
0.4.2

ID:
CVE-2013-4457

LINK

Cocaine Gem for Ruby contains a flaw

DESCRIPTION:

Cocaine Gem for Ruby contains a flaw that is due to the method of variable interpolation used by the program. With a specially crafted object, a context-dependent attacker can execute arbitrary commands.

VULNERABLE GEM: DOORKEEPER@3.1.0

Name:
doorkeeper

Version:
3.1.0

ID:
CVE-2016-6582

LINK

Doorkeeper gem does not revoke tokens & uses wrong auth/auth method

DESCRIPTION:

Doorkeeper failed to implement OAuth 2.0 Token Revocation (RFC 7009) in the following ways:

- **Public clients making valid, unauthenticated calls to revoke a token would not have their token revoked**
- **Requests were not properly authenticating the *client credentials* but were, instead, looking at the access token in a second location**
- **Because of 2, the requests were also not authorizing confidential clients' ability to revoke a given token. It should only revoke tokens that belong to it.**

The security implication is: OAuth 2.0 clients who "log out" a user expect to have the corresponding access & refresh tokens revoked, preventing an attacker who may have already hijacked the session from continuing to impersonate the victim. Because of the bug described above, this is not the case. As far as OWASP is concerned, this counts as broken authentication design.

MITRE has assigned CVE-2016-6582 due to the security issues raised. An attacker, thanks to 1, can replay a hijacked session after a victim logs out/revokes their token. Additionally, thanks to 2 & 3, an attacker via a compromised confidential client could "grief" other clients by revoking their tokens (albeit this is an exceptionally narrow attack with little value).

VULNERABLE GEM: DOORKEEPER@3.1.0

Name:
doorkeeper

Version:
3.1.0

ID:
CVE-2023-34246

LINK

Doorkeeper Improper Authentication vulnerability

DESCRIPTION:

OAuth RFC 8252 says https://www.rfc-editor.org/rfc/rfc8252#section-8.6 the authorization server SHOULD NOT process authorization requests

automatically without user consent or interaction, except when the identity of the client can be assured. **This includes the case where the user has previously approved an authorization request for a given client id**
But Doorkeeper automatically processes authorization requests without user consent for public clients that have been previous approved. Public clients are inherently vulnerable to impersonation, their identity cannot be assured.
Issue https://github.com/doorkeeper-gem/doorkeeper/issues/1589
Fix https://github.com/doorkeeper-gem/doorkeeper/pull/1646

VULNERABLE GEM: DOORKEEPER@3.1.0

Name:
doorkeeper

Version:
3.1.0

ID:
CVE-2018-1000088

LINK

Doorkeeper gem has stored XSS on authorization consent view

DESCRIPTION:

Stored XSS on the OAuth Client's name will cause users being prompted for consent via the "implicit" grant type to execute the XSS payload.
The XSS attack could gain access to the user's active session, resulting in account compromise.
Any user is susceptible if they click the authorization link for the malicious OAuth client. Because of how the links work, a user cannot tell if a link is malicious or not without first visiting the page with the XSS payload.
If 3rd parties are allowed to create OAuth clients in the app using Doorkeeper, upgrade to the patched versions immediately.
Additionally there is stored XSS in the native*redirect*uri form element.
DWF has assigned CVE-2018-1000088.

VULNERABLE GEM: FFI@1.9.3

VULNERABLE GEM: FFI@1.9.3

Name:
ffi

Version:
1.9.3

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: HTTPARTY@0.13.3

Name:
httparty

Version:
0.13.3

ID:
GHSA-5pq7-52mg-hr42

LINK

---

httparty has multipart/form-data request tampering vulnerability

DESCRIPTION:

"multipart/form-data request tampering vulnerability" caused by Content-Disposition "filename" lack of escaping in httparty.
`httparty/lib/httparty/request` > `body.rb` > `def generate_multipart`
https://github.com/jnunemaker/httparty/blob/4416141d37fd71bdba4f37589ec2
By exploiting this problem, the following attacks are possible

An attack that rewrites the \"name\" field according to the crafted file name, impersonating (overwriting) another field.
Attacks that rewrite the filename extension at the time multipart/form-data is generated by tampering with the filename.

VULNERABLE GEM: HTTPARTY@0.13.3

Name:
httparty

Version:
0.13.3

ID:
CVE-2024-22049

LINK

---

httparty has multipart/form-data request tampering vulnerability

DESCRIPTION:

HTTP multipart/form-data request tampering vulnerability in httparty < 0.20.0, due to lack of proper escaping of double quotes within the filename attribute of the Content-Disposition header. If the Content-Disposition header is set to "form-data" and contains the "filename" attribute, and the "filename" attribute contains a double quote followed by additional attributes, then those attributes will be parsed as Content-Disposition attributes and will override the Content-Disposition header's previous attributes.

```
Content-Disposition: form-data; name="avatar"; filename="overwrite_name_field_and_extension.sh"; name="foo"; dummy=".txt"
```

VULNERABLE GEM: JQUERY-RAILS@2.0.2

Name:
jquery-rails

Version:
2.0.2

ID:
CVE-2020-11022

LINK

Potential XSS vulnerability in jQuery

DESCRIPTION:

IMPACT

Passing HTML from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. `.html()`, `.append()`, and others) may execute untrusted code.

PATCHES

This problem is patched in jQuery 3.5.0.

WORKAROUNDS

To workaround the issue without upgrading, adding the following to your code: `js jQuery.htmlPrefilter = function( html ) { return html; };` You need to use at least jQuery 1.12/2.2 or newer to be able to apply this workaround.

REFERENCES

https://blog.jquery.com/2020/04/10/jquery-3-5-0-released/
https://jquery.com/upgrade-guide/3.5/

FOR MORE INFORMATION

If you have any questions or comments about this advisory, search for a relevant issue in the jQuery repo.
If you don't find an answer, open a new issue."

VULNERABLE GEM: JQUERY-RAILS@2.0.2

Name:
jquery-rails

Version:
2.0.2

ID:
CVE-2015-1840

LINK

CSRF Vulnerability in jquery-rails

DESCRIPTION:

In the scenario where an attacker might be able to control the href attribute of an anchor tag or the action attribute of a form tag that will trigger a POST action, the attacker can set the href or action to " https://attacker.com" (note the leading space) that will be passed to JQuery, who will see this as a same origin request, and send the user's CSRF token to the attacker domain.
To work around this problem, change code that allows users to control the href attribute of an anchor tag or the action attribute of a form tag to filter the user parameters.
For example, code like this:
link_to params
to code like this:
link*to filtered*params
def filtered_params # Filter just the parameters that you trust end
See also: - http://blog.honeybadger.io/understanding-the-rails-jquery-csrf-vulnerability-cve-2015-1840/

VULNERABLE GEM: JQUERY-RAILS@2.0.2

Name:
jquery-rails

Version:
2.0.2

ID:
CVE-2020-7656

LINK

Cross-Site Scripting in jquery

DESCRIPTION:

Versions of `jquery` prior to 1.9.0 are vulnerable to Cross-Site Scripting. The load method fails to recognize and remove "", which results in the enclosed script logic to be executed. This allows attackers to execute arbitrary JavaScript in a victim's browser.

VULNERABLE GEM: JQUERY-RAILS@2.0.2

Name:
jquery-rails

Version:
2.0.2

ID:
CVE-2012-6708

LINK

Cross-Site Scripting in jquery

DESCRIPTION:

Affected versions of `jquery` are vulnerable to cross-site scripting. This occurs because the main `jquery` function uses a regular expression to differentiate between HTML and selectors, but does not properly anchor the regular expression. The result is that `jquery` may interpret HTML as selectors when given certain inputs, allowing for client side code execution.

PROOF OF CONCEPT

```
$("#log").html( $("element[attribute=' ']").html() );
```

VULNERABLE GEM: JQUERY-RAILS@2.0.2

Name:
jquery-rails

Version:
2.0.2

ID:

CVE-2019-11358

LINK

Prototype pollution attack through jQuery $.extend

DESCRIPTION:

jQuery before 3.4.0 mishandles jQuery.extend(true, {}, ...) because of bject.prototype pollution. If an unsanitized source object contained an enumerable **proto** property, it could extend the native Object.prototype.

VULNERABLE GEM: JQUERY-RAILS@2.0.2

Name:
jquery-rails

Version:
2.0.2

ID:
CVE-2015-9251

LINK

Cross-Site Scripting (XSS) in jquery

DESCRIPTION:

Affected versions of `jquery` interpret `text/javascript` responses from cross-origin ajax requests, and automatically execute the contents in `jQuery.globalEval`, even when the ajax request doesn't contain the `dataType` option.

VULNERABLE GEM: JQUERY-RAILS@2.0.2

Name:
jquery-rails

Version:
2.0.2

ID:
CVE-2020-11023

LINK

Potential XSS vulnerability in jQuery

DESCRIPTION:

IMPACT

Passing HTML containing `<option>` elements from untrusted sources - even after sanitizing them - to one of jQuery's DOM manipulation methods (i.e. `.html()`, `.append()`, and others) may execute untrusted code.

WORKAROUNDS

To workaround this issue without upgrading, use DOMPurify with its `SAFE_FOR_JQUERY` option to sanitize the HTML string before passing it to a jQuery method.

VULNERABLE GEM: JSON@1.8.6

Name:
json

Version:
1.8.6

ID:
CVE-2020-10663

LINK

---

json Gem for Ruby Unsafe Object Creation Vulnerability (additional fix)

DESCRIPTION:

There is an unsafe object creation vulnerability in the json gem bundled with Ruby. This vulnerability has been assigned the CVE identifier CVE-2020-10663. We strongly recommend upgrading the json gem.

DETAILS

When parsing certain JSON documents, the json gem (including the one bundled with Ruby) can be coerced into creating arbitrary objects in the target system.

This is the same issue as CVE-2013-0269. The previous fix was incomplete, which addressed JSON.parse(user*input), but didnâ€™t address some other styles of JSON parsing including JSON(user*input) and JSON.parse(user_input, nil).

See CVE-2013-0269 in detail. Note that the issue was exploitable to cause a Denial of Service by creating many garbage-uncollectable Symbol objects, but this kind of attack is no longer valid because Symbol objects are now garbage-collectable. However, creating arbitrary objects may cause severe security consequences depending upon the application code.

VULNERABLE GEM: KAMINARI@0.13.0

Name:
kaminari

Version:
0.13.0

ID:
CVE-2024-32978

LINK

Insecure File Permissions vulnerability in kaminari

DESCRIPTION:

kaminari versions prior to 0.16.2 are vulnerable to an Insecure File Permissions vulnerability, where certain files within the kaminari gem have insecure file permissions.

Versions Affected: < 0.16.2 Fixed Versions: >= 0.16.2

Impact

An attacker with local access could write arbitrary code to the affected files resulting in arbitrary code execution.

Releases

The fixed releases are available at the normal locations.

Workarounds

Manually set the permissions of the affected files to `644`.

ALL AFFECTED VERSIONS:

```
lib/kaminari/models/page_scope_methods.rb
```

VERSION 0.15.0 AND 0.15.1:

```
spec/models/mongo_mapper/mongo_mapper_spec.rb
```

VERSION 0.16.0:

```
spec/models/mongo_mapper/mongo_mapper_spec.rb
spec/models/mongoid/mongoid_spec.rb
```

VERSION 0.16.1:

```
spec/models/active_record/scopes_spec.rb
spec/models/mongo_mapper/mongo_mapper_spec.rb
spec/models/mongoid/mongoid_spec.rb
gemfiles/data_mapper_12.gemfile
gemfiles/active_record_32.gemfile
```

VULNERABLE GEM: KAMINARI@0.13.0

Name:
kaminari

Version:
0.13.0

ID:
CVE-2020-11082

LINK

Cross-Site Scripting in Kaminari via `original_script_name` parameter

DESCRIPTION:

IMPACT

There was a vulnerability in versions of Kaminari that would allow an attacker to inject arbitrary code into pages with pagination links.
For example, an attacker could craft pagination links that link to other domain or host: https://example.com/posts?

page=4&original*script*name=https://another-host.example.com
In addition, an attacker could also craft pagination links that include JavaScript code that runs when a user clicks the link:
https://example.com/posts?
page=4&original*script*name=javascript:alert(42)%3b//
RELEASES

The 1.2.1 gem including the patch has already been released. All past released versions are affected by this vulnerability.
WORKAROUNDS

Application developers who can't update the gem can workaround by overriding the `PARAM_KEY_EXCEPT_LIST` constant.

```
module Kaminari::Helpers
PARAM_KEY_EXCEPT_LIST = [:authenticity_token, :commit, :utf8, :_method, :script_name, :original_script_name].freeze
end
```

VULNERABLE GEM: KRAMDOWN@1.4.1

Name:
kramdown

Version:
1.4.1

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: MAIL@2.5.4

Name:
mail

Version:
2.5.4

ID:
CVE-2015-9097

LINK

CVE-2015-9097 rubygem-mail: SMTP injection via recipient email addresses

DESCRIPTION:

The mail gem before 2.5.5 for Ruby (aka A Really Ruby Mail Library) is vulnerable to SMTP command injection via CRLF sequences in a RCPT TO or MAIL FROM command, as demonstrated by CRLF sequences immediately before and after a DATA substring.

VULNERABLE GEM: NOKOGIRI@1.5.5

Name:
nokogiri

Version:
1.5.5

ID:
CVE-2016-4658

LINK

Nokogiri gem contains several vulnerabilities in libxml2 and libxslt

DESCRIPTION:

Nokogiri version 1.7.1 has been released, pulling in several upstream patches to the vendored libxml2 to address the following CVEs:
CVE-2016-4658 CVSS v3 Base Score: 9.8 (Critical) libxml2 in Apple iOS before 10, OS X before 10.12, tvOS before 10, and watchOS before 3 allows remote attackers to execute arbitrary code or cause a denial of service (memory corruption) via a crafted XML document.
CVE-2016-5131 CVSS v3 Base Score: 8.8 (HIGH) Use-after-free vulnerability in libxml2 through 2.9.4, as used in Google Chrome before 52.0.2743.82, allows remote attackers to cause a denial of service or possibly have unspecified other impact via vectors related to the XPointer range-to function.

VULNERABLE GEM: NOKOGIRI@1.5.5

Name:
nokogiri

Version:
1.5.5

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@1.5.5

Name:
nokogiri

Version:
1.5.5

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@1.5.5

Name:
nokogiri

Version:
1.5.5

ID:
CVE-2017-16932

LINK

Nokogiri gem, via libxml, is affected by DoS vulnerabilities

DESCRIPTION:

The version of libxml2 packaged with Nokogiri contains a vulnerability. Nokogiri has mitigated these issue by upgrading to libxml 2.9.5.
Wei Lei discovered that libxml2 incorrecty handled certain parameter entities. An attacker could use this issue with specially constructed XML data to cause libxml2 to consume resources, leading to a denial of service.

VULNERABLE GEM: NOKOGIRI@1.5.5

Name:
nokogiri

Version:
1.5.5

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@1.5.5

Name:
nokogiri

Version:
1.5.5

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@1.5.5

Name:
nokogiri

Version:
1.5.5

ID:
CVE-2017-15412

LINK

Nokogiri gem, via libxml, is affected by DoS vulnerabilities
DESCRIPTION:
The version of libxml2 packaged with Nokogiri contains a vulnerability. Nokogiri has mitigated these issue by upgrading to libxml 2.9.6.
It was discovered that libxml2 incorrecty handled certain files. An attacker could use this issue with specially constructed XML data to cause libxml2 to consume resources, leading to a denial of service.

VULNERABLE GEM: NOKOGIRI@1.5.5

Name:
nokogiri

Version:
1.5.5

ID:
CVE-2013-6461

LINK

CVE-2013-6461 rubygem-nokogiri: DoS while parsing XML entities

DESCRIPTION:

Nokogiri gem 1.5.x and 1.6.x has DoS while parsing XML entities by failing to apply limits

VULNERABLE GEM: NOKOGIRI@1.5.5

Name:
nokogiri

Version:
1.5.5

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@1.5.5

Name:
nokogiri

Version:
1.5.5

ID:
CVE-2017-9050

LINK

Nokogiri gem, via libxml, is affected by DoS and RCE vulnerabilities

DESCRIPTION:

The version of libxml2 packaged with Nokogiri contains several vulnerabilities. Nokogiri has mitigated these issues by upgrading to libxml 2.9.5.
It was discovered that a type confusion error existed in libxml2. An attacker could use this to specially construct XML data that could cause a denial of service or possibly execute arbitrary code. (CVE-2017-0663)
It was discovered that libxml2 did not properly validate parsed entity references. An attacker could use this to specially construct XML data that could expose sensitive information. (CVE-2017-7375)
It was discovered that a buffer overflow existed in libxml2 when handling HTTP redirects. An attacker could use this to specially construct XML data that could cause a denial of service or possibly execute arbitrary code. (CVE-2017-7376)
Marcel BÃ¶hme and Van-Thuan Pham discovered a buffer overflow in

libxml2 when handling elements. An attacker could use this to specially construct XML data that could cause a denial of service or possibly execute arbitrary code. (CVE-2017-9047)
Marcel BÃ¶hme and Van-Thuan Pham discovered a buffer overread in libxml2 when handling elements. An attacker could use this to specially construct XML data that could cause a denial of service. (CVE-2017-9048)
Marcel BÃ¶hme and Van-Thuan Pham discovered multiple buffer overreads in libxml2 when handling parameter-entity references. An attacker could use these to specially construct XML data that could cause a denial of service. (CVE-2017-9049, CVE-2017-9050)

VULNERABLE GEM: NOKOGIRI@1.5.5

Name:
nokogiri

Version:
1.5.5

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@1.5.5

Name:
nokogiri

Version:
1.5.5

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

Severity: High

Type: CWE-787 Out of bounds write

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@1.5.5

Name:
nokogiri

Version:
1.5.5

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@1.5.5

Name:
nokogiri

Version:
1.5.5

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@1.5.5

Name:
nokogiri

Version:
1.5.5

ID:
CVE-2017-18258

LINK

Moderate severity vulnerability that affects nokogiri

DESCRIPTION:

The xz_head function in xzlib.c in libxml2 before 2.9.6 allows remote attackers to cause a denial of service (memory consumption) via a crafted LZMA file, because the decoder functionality does not restrict memory usage to what is required for a legitimate file.

References: - https://nvd.nist.gov/vuln/detail/CVE-2017-18258 - https://git.gnome.org/browse/libxml2/commit/?id=e2a9122b8dde53d320750451e9907a7dcb2ca8bb - https://github.com/advisories/GHSA-882p-jqgm-f45g - https://kc.mcafee.com/corporate/index?page=content&id=SB10284 - https://lists.debian.org/debian-lts-announce/2018/09/msg00035.html - https://lists.debian.org/debian-lts-announce/2020/09/msg00009.html - https://security.netapp.com/advisory/ntap-20190719-0001/ - https://usn.ubuntu.com/3739-1/

VULNERABLE GEM: NOKOGIRI@1.5.5

Name:
nokogiri

Version:
1.5.5

ID:
CVE-2017-5029

LINK

Nokogiri gem contains two upstream vulnerabilities in libxslt 1.1.29

DESCRIPTION:

nokogiri version 1.7.2 has been released.
This is a security update based on 1.7.1, addressing two upstream libxslt 1.1.29 vulnerabilities classified as "Medium" by Canonical and given a CVSS3 score of "6.5 Medium" and "8.8 High" by RedHat.
These patches only apply when using Nokogiri's vendored libxslt package. If you're using your distro's system libraries, there's no need to upgrade from 1.7.0.1 or 1.7.1 at this time.
Full details are available at the github issue linked to in the changelog below.

1.7.2 / 2017-05-09

SECURITY NOTES

[MRI] Upstream libxslt patches are applied to the vendored libxslt 1.1.29 which address CVE-2017-5029 and CVE-2016-4738.
For more information:

https://github.com/sparklemotion/nokogiri/issues/1634
http://people.canonical.com/~ubuntu-security/cve/2017/CVE-2017-5029.html
http://people.canonical.com/~ubuntu-security/cve/2016/CVE-2016-4738.html

VULNERABLE GEM: NOKOGIRI@1.5.5

Name:
nokogiri

Version:
1.5.5

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@1.5.5

Name:
nokogiri

Version:
1.5.5

ID:
CVE-2015-1819

LINK

Nokogiri gem contains several vulnerabilities in libxml2 and libxslt

DESCRIPTION:

Several vulnerabilities were discovered in the libxml2 and libxslt libraries that the Nokogiri gem depends on.
CVE-2015-1819 A denial of service flaw was found in the way libxml2 parsed XML documents. This flaw could cause an application that uses libxml2 to use an excessive amount of memory.
CVE-2015-7941 libxml2 does not properly stop parsing invalid input, which allows context-dependent attackers to cause a denial of service (out-of-

bounds read and libxml2 crash) via crafted specially XML data.
CVE-2015-7942 The xmlParseConditionalSections function in parser.c in libxml2 does not properly skip intermediary entities when it stops parsing invalid input, which allows context-dependent attackers to cause a denial of service (out-of-bounds read and crash) via crafted XML data.
CVE-2015-7995 The xsltStylePreCompute function in preproc.c in libxslt 1.1.28 does not check whether the parent node is an element, which allows attackers to cause a denial of service using a specially crafted XML document.
CVE-2015-8035 The xz_decomp function in xzlib.c in libxml2 2.9.1 does not properly detect compression errors, which allows context-dependent attackers to cause a denial of service (process hang) via crafted XML data.
Another vulnerability was discoverd in libxml2 that could cause parsing of unclosed comments to result in "conditional jump or move depends on uninitialized value(s)" and unsafe memory access. This issue does not have a CVE assigned yet. See related URLs for details. Patched in v1.6.7.rc4.

VULNERABLE GEM: NOKOGIRI@1.5.5

Name:
nokogiri

Version:
1.5.5

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@1.5.5

Name:
nokogiri

Version:
1.5.5

ID:
OSVDB-118481

LINK

Nokogiri Gem for JRuby XML Document Root Element Handling Memory Consumption Remote DoS

DESCRIPTION:

Nokogiri Gem for JRuby contains a flaw that is triggered when handling a root element in an XML document. This may allow a remote attacker to cause a consumption of memory resources.

VULNERABLE GEM: NOKOGIRI@1.5.5

Name:
nokogiri

Version:
1.5.5

ID:
GHSA-r95h-9x8f-r3f7

LINK

Nokogiri updates packaged libxml2 to v2.12.7 to resolve CVE-2024-34459

DESCRIPTION:

SUMMARY

Nokogiri v1.16.5 upgrades its dependency libxml2 to 2.12.7 from 2.12.6. libxml2 v2.12.7 addresses CVE-2024-34459:

- **described at https://gitlab.gnome.org/GNOME/libxml2/-/issues/720**
- **patched by https://gitlab.gnome.org/GNOME/libxml2/-/commit/2876ac53**

IMPACT

There is no impact to Nokogiri users because the issue is present only in libxml2's `xmllint` tool which Nokogiri does not provide or expose.

TIMELINE

- **2024-05-13 05:57 EDT, libxml2 2.12.7 release is announced**
- **2024-05-13 08:30 EDT, nokogiri maintainers begin triage**
- **2024-05-13 10:05 EDT, nokogiri v1.16.5 is released and this GHSA made public**

VULNERABLE GEM: NOKOGIRI@1.5.5

Name:
nokogiri

Version:
1.5.5

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@1.5.5

Name:
nokogiri

Version:
1.5.5

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: **Type: Arbitrary code execution**
Medium
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@1.5.5

Name:
nokogiri

Version:
1.5.5

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@1.5.5

Name:
nokogiri

Version:
1.5.5

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 accessed over the network
# but in v1.11.0.rc4 and later, this call will disallow network access for external resources
schema = Nokogiri::XML::Schema.new(schema)

# in v1.11.0.rc4 and later, the following is equivalent to the code above
# (the second parameter is optional, and this demonstrates its default value)
schema = Nokogiri::XML::Schema.new(schema, Nokogiri::XML::ParseOptions::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@1.5.5

Name:
nokogiri

Version:
1.5.5

ID:
GHSA-mrxw-mxhj-p664

LINK

Nokogiri updates packaged libxslt to v1.1.43 to resolve multiple CVEs

DESCRIPTION:

SUMMARY

Nokogiri v1.18.4 upgrades its dependency libxslt to v1.1.43.
libxslt v1.1.43 resolves:

- **CVE-2025-24855: Fix use-after-free of XPath context node**
- **CVE-2024-55549: Fix UAF related to excluded namespaces**

IMPACT

CVE-2025-24855

- **"Use-after-free due to xsltEvalXPathStringNs leaking xpathCtxt->node"**
- **MITRE has rated this 7.8 High**
- **CVSS:3.1/AV:L/AC:H/PR:N/UI:N/S:C/C:N/I:H/A:H**
- **Upstream report: https://gitlab.gnome.org/GNOME/libxslt/-/issues/128**
- **NVD entry: https://nvd.nist.gov/vuln/detail/CVE-2025-24855**

CVE-2024-55549

- **"Use-after-free related to excluded result prefixes"**
- **MITRE has rated this 7.8 High**
- **CVSS:3.1/AV:L/AC:H/PR:N/UI:N/S:C/C:N/I:H/A:H**
- **Upstream report: https://gitlab.gnome.org/GNOME/libxslt/-/issues/127**

NVD entry: https://nvd.nist.gov/vuln/detail/CVE-2024-55549

VULNERABLE GEM: NOKOGIRI@1.5.5

Name:
nokogiri

Version:
1.5.5

ID:
CVE-2013-6460

LINK

CVE-2013-6460 rubygem-nokogiri: DoS while parsing XML documents

DESCRIPTION:

Nokogiri gem 1.5.x has Denial of Service via infinite loop when parsing XML documents

VULNERABLE GEM: NOKOGIRI@1.5.5

Name:
nokogiri

Version:
1.5.5

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@1.5.5

Name:
nokogiri

Version:
1.5.5

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@1.5.5

Name:
nokogiri

Version:
1.5.5

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@1.5.5

Name:
nokogiri

Version:
1.5.5

ID:
GHSA-vvfq-8hwr-qm4m

LINK

Nokogiri updates packaged libxml2 to 2.13.6 to resolve CVE-2025-24928 and CVE-2024-56171
DESCRIPTION:
SUMMARY
Nokogiri v1.18.3 upgrades its dependency libxml2 to v2.13.6.
libxml2 v2.13.6 addresses:
CVE-2025-24928
described at
https://gitlab.gnome.org/GNOME/libxml2/-/issues/847

CVE-2024-56171
described at
https://gitlab.gnome.org/GNOME/libxml2/-/issues/828

IMPACT
CVE-2025-24928

Stack-buffer overflow is possible when reporting DTD validation errors if the input contains a long (~3kb) QName prefix.
CVE-2024-56171

Use-after-free is possible during validation against untrusted XML Schemas (.xsd) and, potentially, validation of untrusted documents against trusted Schemas if they make use of `xsd:keyref` in combination with recursively defined types that have additional identity constraints.

VULNERABLE GEM: NOKOGIRI@1.5.5

Name:
nokogiri

Version:
1.5.5

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: NOKOGIRI@1.5.5

Name:
nokogiri

Version:
1.5.5

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@1.5.5

Name:	Version:
nokogiri	1.5.5

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@1.5.5

Name:
nokogiri

Version:
1.5.5

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@1.5.5

Name:
nokogiri

Version:
1.5.5

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@1.5.5

Name:
nokogiri

Version:
1.5.5

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@1.5.5

Name:
nokogiri

Version:
1.5.5

ID:
GHSA-5w6v-399v-w3cc

LINK

Nokogiri updates packaged libxml2 to v2.13.8 to resolve CVE-2025-32414 and CVE-2025-32415

DESCRIPTION:

SUMMARY

Nokogiri v1.18.8 upgrades its dependency libxml2 to v2.13.8.
libxml2 v2.13.8 addresses:

CVE-2025-32414
described at
https://gitlab.gnome.org/GNOME/libxml2/-/issues/889

CVE-2025-32415
described at
https://gitlab.gnome.org/GNOME/libxml2/-/issues/890

IMPACT

CVE-2025-32414: NO IMPACT

In libxml2 before 2.13.8 and 2.14.x before 2.14.2, out-of-bounds memory access can occur in the Python API (Python bindings) because of an incorrect return value. This occurs in xmlPythonFileRead and xmlPythonFileReadRaw because of a difference between bytes and characters.
There is no impact from this CVE for Nokogiri users.
CVE-2025-32415: LOW IMPACT

In libxml2 before 2.13.8 and 2.14.x before 2.14.2, xmlSchemaIDCFillNodeTables in xmlschemas.c has a heap-based buffer under-read. To exploit this, a crafted XML document must be validated against an XML schema with certain identity constraints, or a crafted XML schema must be used.
In the upstream issue, further context is provided by the maintainer:
The bug affects validation against untrusted XML Schemas (.xsd) and validation of untrusted documents against trusted Schemas if they make use of xsd:keyref in combination with recursively defined types that have additional identity constraints.
MITRE has published a severity score of 2.9 LOW (CVSS:3.1/AV:L/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:L) for this CVE.

VULNERABLE GEM: NOKOGIRI@1.5.5

Name:
nokogiri

Version:
1.5.5

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: OAUTH@0.4.6

Name:
oauth

Version:
0.4.6

ID:
CVE-2016-11086

LINK

Improper Certificate Validation in oauth ruby gem

Improper Certificate Validation in oauth ruby gem

DESCRIPTION:

lib/oauth/consumer.rb in the oauth-ruby gem through 0.5.4 for Ruby does not verify server X.509 certificates if a certificate bundle cannot be found, which allows man-in-the-middle attackers to spoof servers and obtain sensitive information.

VULNERABLE GEM: PAPERCLIP@3.3.1

Name:
paperclip

Version:
3.3.1

ID:
OSVDB-103151

LINK

Paperclip: Access Restriction Bypass

DESCRIPTION:

Paperclip Gem for Ruby contains a flaw that is due to the application failing to properly validate the file extension, instead only validating the Content-Type header during file uploads. This may allow a remote attacker to bypass restrictions on file types for uploaded files by spoofing the content-type.

VULNERABLE GEM: PAPERCLIP@3.3.1

VULNERABLE GEM: PAPERCLIP@3.3.1

Name:
paperclip

Version:
3.3.1

ID:
CVE-2015-2963

LINK

Paperclip Gem for Ruby vulnerable to content type spoofing

DESCRIPTION:

There is an issue where if an HTML file is uploaded with a .html extension, but the content type is listed as being `image/jpeg`, this will bypass a validation checking for images. But it will also pass the spoof check, because a file named .html and containing actual HTML passes the spoof check.

VULNERABLE GEM: PAPERCLIP@3.3.1

Name:
paperclip

Version:
3.3.1

ID:
CVE-2017-0889

LINK

Paperclip ruby gem suffers from a Server-Side Request Forgery (SSRF) vulnerability in the Paperclip::UriAdapter and Paperclip::HttpUrlProxyAdapter class.

DESCRIPTION:

Paperclip gem provides multiple ways a file can be uploaded to a web server. The vulnerability affects two of Paperclipâ€™s IO adapters that accept URLs as attachment data (UriAdapter and HttpUrlProxyAdapter). When these adapters are used, Paperclip acts as a proxy and downloads the file from the website URI that is passed in. The library does not perform

any validation to protect against Server Side Request Forgery (SSRF) exploits by default. This may allow a remote attacker to access information about internal network resources.

VULNERABLE GEM: RACK@1.4.7

Name:
rack

Version:
1.4.7

ID:
CVE-2022-30122

LINK

Denial of Service Vulnerability in Rack Multipart Parsing

DESCRIPTION:

There is a possible denial of service vulnerability in the multipart parsing component of Rack. This vulnerability has been assigned the CVE identifier CVE-2022-30122.

Versions Affected: >= 1.2 Not affected: < 1.2 Fixed Versions: 2.0.9.1, 2.1.4.1, 2.2.3.1

IMPACT

Carefully crafted multipart POST requests can cause Rack's multipart parser to take much longer than expected, leading to a possible denial of service vulnerability.

Impacted code will use Rack's multipart parser to parse multipart posts. This includes directly using the multipart parser like this:

```
params = Rack::Multipart.parse_multipart(env)
```

But it also includes reading POST data from a Rack request object like this:

```
p request.POST # read POST data
p request.params # reads both query params and POST data
```

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.

VULNERABLE GEM: RACK@1.4.7

Name:
rack

Version:
1.4.7

ID:
CVE-2022-30123

LINK

Possible shell escape sequence injection vulnerability in Rack

DESCRIPTION:

There is a possible shell escape sequence injection vulnerability in the Lint and CommonLogger components of Rack. This vulnerability has been assigned the CVE identifier CVE-2022-30123.
Versions Affected: All. Not affected: None Fixed Versions: 2.0.9.1, 2.1.4.1, 2.2.3.1

IMPACT

Carefully crafted requests can cause shell escape sequences to be written to the terminal via Rack's Lint middleware and CommonLogger middleware. These escape sequences can be leveraged to possibly execute commands in the victim's terminal.
Impacted applications will have either of these middleware installed, and vulnerable apps may have something like this:

```
use Rack::Lint
```

Or

```
use Rack::CommonLogger
```

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

WORKAROUNDS

Remove these middleware from your application

VULNERABLE GEM: RACK@1.4.7

Name:
rack

Version:
1.4.7

ID:
CVE-2018-16471

LINK

Possible XSS vulnerability in Rack

DESCRIPTION:

There is a possible vulnerability in Rack. This vulnerability has been assigned the CVE identifier CVE-2018-16471.
Versions Affected: All. Not affected: None. Fixed Versions: 2.0.6, 1.6.11

IMPACT

There is a possible XSS vulnerability in Rack. Carefully crafted requests can impact the data returned by the `scheme` method on `Rack::Request`. Applications that expect the scheme to be limited to "http" or "https" and do not escape the return value could be vulnerable to an XSS attack.
Vulnerable code looks something like this:

```
<%= request.scheme.html_safe %>
```

Note that applications using the normal escaping mechanisms provided by Rails may not impacted, but applications that bypass the escaping mechanisms, or do not use them may be vulnerable.
All users running an affected release should either upgrade or use one of the workarounds immediately.

RELEASES

The 2.0.6 and 1.6.11 releases are available at the normal locations.

WORKAROUNDS

The following monkey patch can be applied to work around this issue:

```
  require "rack"
require "rack/request"

class Rack::Request
SCHEME_WHITELIST = %w(https http).freeze

def scheme
  if get_header(Rack::HTTPS) == 'on'
    'https'
  elsif get_header(HTTP_X_FORWARDED_SSL) == 'on'
    'https'
  elsif forwarded_scheme
    forwarded_scheme
  else
    get_header(Rack::RACK_URL_SCHEME)
  end
end

def forwarded_scheme
  scheme_headers = [
    get_header(HTTP_X_FORWARDED_SCHEME),
    get_header(HTTP_X_FORWARDED_PROTO).to_s.split(',')[0]
  ]

  scheme_headers.each do |header|
    return header if SCHEME_WHITELIST.include?(header)
  end

  nil
end
end
```

VULNERABLE GEM: RACK@1.4.7

Name:	Version:
rack	1.4.7

ID:
CVE-2024-26146

LINK

Possible Denial of Service Vulnerability in Rack Header Parsing

DESCRIPTION:

There is a possible denial of service vulnerability in the header parsing routines in Rack. This vulnerability has been assigned the CVE identifier CVE-2024-26146.

Versions Affected: All. Not affected: None Fixed Versions: 2.0.9.4, 2.1.4.4, 2.2.8.1, 3.0.9.1

Impact

Carefully crafted headers can cause header parsing in Rack to take longer than expected resulting in a possible denial of service issue. `Accept` and `Forwarded` headers are impacted.

Ruby 3.2 has mitigations for this problem, so Rack applications using Ruby 3.2 or newer are unaffected.

Releases

The fixed releases are available at the normal locations.

Workarounds

There are no feasible workarounds for this issue.

VULNERABLE GEM: RACK@1.4.7

Name:
rack

Version:
1.4.7

ID:
CVE-2023-27530

LINK

Possible DoS Vulnerability in Multipart MIME parsing

DESCRIPTION:

There is a possible DoS vulnerability in the Multipart MIME parsing code in Rack. This vulnerability has been assigned the CVE identifier CVE-2023-27530.

Versions Affected: All. Not affected: None Fixed Versions: 3.0.4.2, 2.2.6.3,

2.1.4.3, 2.0.9.3

Impact

The Multipart MIME parsing code in Rack limits the number of file parts, but does not limit the total number of parts that can be uploaded. Carefully crafted requests can abuse this and cause multipart parsing to take longer than expected.

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

Workarounds

A proxy can be configured to limit the POST body size which will mitigate this issue.

VULNERABLE GEM: RACK@1.4.7

Name:
rack

Version:
1.4.7

ID:
CVE-2020-8184

LINK

Percent-encoded cookies can be used to overwrite existing prefixed cookie names

DESCRIPTION:

It is possible to forge a secure or host-only cookie prefix in Rack using an arbitrary cookie write by using URL encoding (percent-encoding) on the name of the cookie. This could result in an application that is dependent on this prefix to determine if a cookie is safe to process being manipulated into processing an insecure or cross-origin request. This vulnerability has been assigned the CVE identifier CVE-2020-8184.

Versions Affected: rack < 2.2.3, rack < 2.1.4 Not affected: Applications which do not rely on *_Host- and* _Secure- prefixes to determine if a cookie is safe to process Fixed Versions: rack >= 2.2.3, rack >= 2.1.4

IMPACT

An attacker may be able to trick a vulnerable application into processing an insecure (non-SSL) or cross-origin request if they can gain the ability to write arbitrary cookies that are sent to the application.

WORKAROUNDS

If your application is impacted but you cannot upgrade to the released versions or apply the provided patch, this issue can be temporarily

addressed by adding the following workaround:

```
 module Rack
 module Utils
   module_function def parse_cookies_header(header)
     return {} unless header
     header.split(/[;] */n).each_with_object({}) do |cookie, cookies|
       next if cookie.empty?
       key, value = cookie.split('=', 2)
       cookies[key] = (unescape(value) rescue value) unless cookies
key?(key)
     end
   end
 end
end
```

VULNERABLE GEM: RACK@1.4.7

Name:
rack

Version:
1.4.7

ID:
CVE-2025-27610

LINK

Local File Inclusion in Rack::Static

DESCRIPTION:

SUMMARY

`Rack::Static` can serve files under the specified `root:` even if `urls:` are provided, which may expose other files under the specified `root:` unexpectedly.

DETAILS

The vulnerability occurs because `Rack::Static` does not properly sanitize user-supplied paths before serving files. Specifically, encoded path traversal sequences are not correctly validated, allowing attackers to access files outside the designated static file directory.

IMPACT

By exploiting this vulnerability, an attacker can gain access to all files under

the specified `root:` directory, provided they are able to determine then path of the file.

MITIGATION

Update to the latest version of Rack,
or
Remove usage of `Rack::Static`,
or
Ensure that `root:` points at a directory path which only contains files which should be accessed publicly.

It is likely that a CDN or similar static file server would also mitigate the issue.

VULNERABLE GEM: RACK@1.4.7

Name:
rack

Version:
1.4.7

ID:
CVE-2025-25184

LINK

Possible Log Injection in Rack::CommonLogger

DESCRIPTION:

SUMMARY

`Rack::CommonLogger` can be exploited by crafting input that includes newline characters to manipulate log entries. The supplied proof-of-concept demonstrates injecting malicious content into logs.

DETAILS

When a user provides the authorization credentials via `Rack::Auth::Basic`, if success, the username will be put in `env['REMOTE_USER']` and later be used by `Rack::CommonLogger` for logging purposes.
The issue occurs when a server intentionally or unintentionally allows a user creation with the username contain CRLF and white space characters, or the server just want to log every login attempts. If an attacker enters a username with CRLF character, the logger will log the malicious username with CRLF characters into the logfile.

IMPACT

Attackers can break log formats or insert fraudulent entries, potentially

obscuring real activity or injecting malicious data into log files.

MITIGATION

Update to the latest version of Rack.

VULNERABLE GEM: RACK@1.4.7

Name:
rack

Version:
1.4.7

ID:
CVE-2022-44572

LINK

Denial of service via multipart parsing in Rack

DESCRIPTION:

There is a denial of service vulnerability in the multipart parsing component of Rack. This vulnerability has been assigned the CVE identifier CVE-2022-44572.

Versions Affected: >= 2.0.0 Not affected: None. Fixed Versions: 2.0.9.2, 2.1.4.2, 2.2.6.1, 3.0.4.1

Impact

Carefully crafted input can cause RFC2183 multipart boundary parsing in Rack to take an unexpected amount of time, possibly resulting in a denial of service attack vector. Any applications that parse multipart posts using Rack (virtually all Rails applications) are impacted.

Workarounds

There are no feasible workarounds for this issue.

VULNERABLE GEM: RACK@1.4.7

VULNERABLE GEM: RACK@1.4.7

Name:
rack

Version:
1.4.7

ID:
CVE-2022-44570

LINK

Denial of service via header parsing in Rack

DESCRIPTION:

There is a possible denial of service vulnerability in the Range header parsing component of Rack. This vulnerability has been assigned the CVE identifier CVE-2022-44570.

Versions Affected: >= 1.5.0 Not affected: None. Fixed Versions: 2.0.9.2, 2.1.4.2, 2.2.6.2, 3.0.4.1

Impact

Carefully crafted input can cause the Range header parsing component in Rack to take an unexpected amount of time, possibly resulting in a denial of service attack vector. Any applications that deal with Range requests (such as streaming applications, or applications that serve files) may be impacted.

Workarounds

There are no feasible workarounds for this issue.

VULNERABLE GEM: RACK@1.4.7

Name:
rack

Version:
1.4.7

ID:
CVE-2022-44571

LINK

Denial of Service Vulnerability in Rack Content-Disposition parsing

DESCRIPTION:

There is a denial of service vulnerability in the Content-Disposition parsing component of Rack. This vulnerability has been assigned the CVE identifier CVE-2022-44571.

Versions Affected: >= 2.0.0 Not affected: None. Fixed Versions: 2.0.9.2, 2.1.4.2, 2.2.6.1, 3.0.4.1

Impact

Carefully crafted input can cause Content-Disposition header parsing in Rack to take an unexpected amount of time, possibly resulting in a denial of service attack vector. This header is used typically used in multipart parsing. Any applications that parse multipart posts using Rack (virtually all Rails applications) are impacted.

Workarounds

There are no feasible workarounds for this issue.

VULNERABLE GEM: RACK@1.4.7

Name:
rack

Version:
1.4.7

ID:
CVE-2025-46727

LINK

Rack has an Unbounded-Parameter DoS in Rack::QueryParser

DESCRIPTION:

SUMMARY

`Rack::QueryParser` parses query strings and `application/x-www-form-urlencoded` bodies into Ruby data structures without imposing any limit on the number of parameters, allowing attackers to send requests with extremely large numbers of parameters.

DETAILS

The vulnerability arises because `Rack::QueryParser` iterates over each `&`-separated key-value pair and adds it to a Hash without enforcing an upper bound on the total number of parameters. This allows an attacker to send a single request containing hundreds of thousands (or more) of parameters, which consumes excessive memory and CPU during parsing.

IMPACT

An attacker can trigger denial of service by sending specifically crafted HTTP requests, which can cause memory exhaustion or pin CPU resources, stalling or crashing the Rack server. This results in full service disruption until the affected worker is restarted.

MITIGATION

Update to a version of Rack that limits the number of parameters parsed, or
Use middleware to enforce a maximum query string size or parameter count, or
Employ a reverse proxy (such as Nginx) to limit request sizes and reject oversized query strings or bodies.

Limiting request body sizes and query string lengths at the web server or CDN level is an effective mitigation.

VULNERABLE GEM: RACK@1.4.7

Name:
rack

Version:
1.4.7

ID:
CVE-2024-25126

LINK

Denial of Service Vulnerability in Rack Content-Type Parsing

DESCRIPTION:

There is a possible denial of service vulnerability in the content type parsing component of Rack. This vulnerability has been assigned the CVE identifier CVE-2024-25126.

Versions Affected: >= 0.4 Not affected: < 0.4 Fixed Versions: 3.0.9.1, 2.2.8.1

Impact

Carefully crafted content type headers can cause Rackâ€™s media type parser to take much longer than expected, leading to a possible denial of service vulnerability.

Impacted code will use Rackâ€™s media type parser to parse content type headers. This code will look like below:

```
request.media_type

## OR
request.media_type_params

## OR
Rack::MediaType.type(content_type)
```

Some frameworks (including Rails) call this code internally, so upgrading is recommended!
All users running an affected release should either upgrade or use one of the workarounds immediately.

Releases

The fixed releases are available at the normal locations.

Workarounds

There are no feasible workarounds for this issue.

VULNERABLE GEM: RACK@1.4.7

Name:
rack

Version:
1.4.7

ID:
CVE-2020-8161

LINK

Directory traversal in Rack::Directory app bundled with Rack

DESCRIPTION:

There was a possible directory traversal vulnerability in the Rack::Directory app that is bundled with Rack.
Versions Affected: rack < 2.2.0 Not affected: Applications that do not use Rack::Directory. Fixed Versions: 2.1.3, >= 2.2.0

IMPACT

If certain directories exist in a director that is managed by `Rack::Directory`, an attacker could, using this vulnerability, read the contents of files on the server that were outside of the root specified in the Rack::Directory initializer.

WORKAROUNDS

Until such time as the patch is applied or their Rack version is upgraded, we recommend that developers do not use Rack::Directory in their applications.

VULNERABLE GEM: RACK@1.4.7

Name:
rack

Version:
1.4.7

ID:
CVE-2019-16782

LINK

Possible information leak / session hijack vulnerability

DESCRIPTION:

There's a possible information leak / session hijack vulnerability in Rack. Attackers may be able to find and hijack sessions by using timing attacks targeting the session id. Session ids are usually stored and indexed in a database that uses some kind of scheme for speeding up lookups of that session id. By carefully measuring the amount of time it takes to look up a session, an attacker may be able to find a valid session id and hijack the session.

The session id itself may be generated randomly, but the way the session is indexed by the backing store does not use a secure comparison.

Impact:

The session id stored in a cookie is the same id that is used when querying the backing session storage engine. Most storage mechanisms (for example a database) use some sort of indexing in order to speed up the lookup of that id. By carefully timing requests and session lookup failures, an attacker may be able to perform a timing attack to determine an existing session id and hijack that session.

VULNERABLE GEM: RACK@1.4.7

Name:
rack

Version:
1.4.7

ID:
CVE-2024-26141

LINK

Possible DoS Vulnerability with Range Header in Rack

DESCRIPTION:

There is a possible DoS vulnerability relating to the Range request header in Rack. This vulnerability has been assigned the CVE identifier CVE-2024-26141.

Versions Affected: >= 1.3.0. Not affected: < 1.3.0 Fixed Versions: 3.0.9.1, 2.2.8.1

Impact

Carefully crafted Range headers can cause a server to respond with an unexpectedly large response. Responding with such large responses could lead to a denial of service issue.

Vulnerable applications will use the `Rack::File` middleware or the `Rack::Utils.byte_ranges` methods (this includes Rails applications).

Releases

The fixed releases are available at the normal locations.

Workarounds

There are no feasible workarounds for this issue.

VULNERABLE GEM: RACK@1.4.7

Name:
rack

Version:
1.4.7

ID:
CVE-2025-32441

LINK

Rack session gets restored after deletion

DESCRIPTION:

SUMMARY

When using the `Rack::Session::Pool` middleware, simultaneous rack requests can restore a deleted rack session, which allows the unauthenticated user to occupy that session.

DETAILS

Rack session middleware prepares the session at the beginning of request, then saves is back to the store with possible changes applied by host rack application. This way the session becomes to be a subject of race conditions

in general sense over concurrent rack requests.
IMPACT

When using the `Rack::Session::Pool` middleware, and provided the attacker can acquire a session cookie (already a major issue), the session may be restored if the attacker can trigger a long running request (within that same session) adjacent to the user logging out, in order to retain illicit access even after a user has attempted to logout.

MITIGATION

Update to the latest version of `rack`,
or
Ensure your application invalidates sessions atomically by marking them as logged out e.g., using a `logged_out` flag, instead of deleting them, and check this flag on every request to prevent reuse, or
Implement a custom session store that tracks session invalidation timestamps and refuses to accept session data if the session was invalidated after the request began.

RELATED

As this code was moved to `rack-session` in Rack 3+, see https://github.com/rack/rack-session/security/advisories/GHSA-9j94-67jr-4cqj for the equivalent advisory in `rack-session` (affecting Rack 3+ only).

VULNERABLE GEM: RACK@1.4.7

Name:
rack

Version:
1.4.7

ID:
CVE-2023-27539

LINK

Possible Denial of Service Vulnerability in Rackâ€™s header parsing
DESCRIPTION:

There is a denial of service vulnerability in the header parsing component of Rack. This vulnerability has been assigned the CVE identifier CVE-2023-27539.
Versions Affected: >= 2.0.0 Not affected: None. Fixed Versions: 2.2.6.4, 3.0.6.1

Impact

Carefully crafted input can cause header parsing in Rack to take an unexpected amount of time, possibly resulting in a denial of service attack vector. Any applications that parse headers using Rack (virtually all Rails applications) are impacted.

Workarounds

Setting Regexp.timeout in Ruby 3.2 is a possible workaround.

VULNERABLE GEM: RACK@1.4.7

Name:
rack

Version:
1.4.7

ID:
CVE-2025-27111

LINK

Escape Sequence Injection vulnerability in Rack lead to Possible Log Injection

DESCRIPTION:

SUMMARY

`Rack::Sendfile` can be exploited by crafting input that includes newline characters to manipulate log entries.

DETAILS

The `Rack::Sendfile` middleware logs unsanitized header values from the `X-Sendfile-Type` header. An attacker can exploit this by injecting escape sequences (such as newline characters) into the header, resulting in log injection.

IMPACT

This vulnerability can distort log files, obscure attack traces, and complicate security auditing.

MITIGATION

Update to the latest version of Rack,

or

Remove usage of `Rack::Sendfile`.

VULNERABLE GEM: RACK-CORS@1.0.2

Name:
rack-cors

Version:
1.0.2

ID:
CVE-2019-18978

LINK

---

rack-cors directory traversal via path

DESCRIPTION:

An issue was discovered in the rack-cors (aka Rack CORS Middleware) gem before 1.0.4 for Ruby. It allows ../ directory traversal to access private resources because resource matching does not ensure that pathnames are in a canonical format.

VULNERABLE GEM: RAKE@10.5.0

Name:
rake

Version:
10.5.0

ID:

CVE-2020-8130

LINK

OS Command Injection in Rake

DESCRIPTION:

There is an OS command injection vulnerability in Ruby Rake < 12.3.3 in Rake::FileList when supplying a filename that begins with the pipe character `|`.

VULNERABLE GEM: RDOC@3.12.2

Name:
rdoc

Version:
3.12.2

ID:
CVE-2024-27281

LINK

RCE vulnerability with .rdoc_options in RDoc

DESCRIPTION:

An issue was discovered in RDoc 6.3.3 through 6.6.2, as distributed in Ruby 3.x through 3.3.0.
When parsing `.rdoc_options` (used for configuration in RDoc) as a YAML file, object injection and resultant remote code execution are possible because there are no restrictions on the classes that can be restored.
When loading the documentation cache, object injection and resultant remote code execution are also possible if there were a crafted cache.
We recommend to update the RDoc gem to version 6.6.3.1 or later. In order to ensure compatibility with bundled version in older Ruby series, you may update as follows instead:

For Ruby 3.0 users: Update to `rdoc` 6.3.4.1

For Ruby 3.1 users: Update to `rdoc` 6.4.1.1
For Ruby 3.2 users: Update to `rdoc` 6.5.1.1

You can use `gem update rdoc` to update it. If you are using bundler, please add `gem "rdoc", ">= 6.6.3.1"` to your `Gemfile`.
Note: 6.3.4, 6.4.1, 6.5.1 and 6.6.3 have a incorrect fix. We recommend to upgrade 6.3.4.1, 6.4.1.1, 6.5.1.1 and 6.6.3.1 instead of them.

VULNERABLE GEM: RDOC@3.12.2

Name:
rdoc

Version:
3.12.2

ID:
CVE-2021-31799

LINK

RDoc OS command injection vulnerability

DESCRIPTION:

RDoc used to call `Kernel#open` to open a local file. If a Ruby project has a file whose name starts with `|` and ends with `tags`, the command following the pipe character is executed. A malicious Ruby project could exploit it to run an arbitrary command execution against a user who attempts to run `rdoc` command.

VULNERABLE GEM: RMAGICK@2.13.2

VULNERABLE GEM: RMAGICK@2.13.2

Name:
rmagick

Version:
2.13.2

ID:
CVE-2023-5349

LINK

---

memory leak flaw was found in ruby-magick

DESCRIPTION:

A memory leak flaw was found in ruby-magick, an interface between Ruby and ImageMagick. This issue can lead to a denial of service (DOS) by memory exhaustion.

VULNERABLE GEM: RUBYZIP@0.9.9

Name:
rubyzip

Version:
0.9.9

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: RUBYZIP@0.9.9

Name:
rubyzip

Version:
0.9.9

ID:
CVE-2017-5946

LINK

Directory traversal vulnerability in rubyzip

DESCRIPTION:

The Zip::File component in the rubyzip gem before 1.2.1 for Ruby has a directory traversal vulnerability. If a site allows uploading of .zip files, an attacker can upload a malicious file that uses "../" pathname substrings to write arbitrary files to the filesystem.

VULNERABLE GEM: RUBYZIP@0.9.9

Name:
rubyzip

Version:
0.9.9

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: SIMPLE_FORM@2.0.1

Name:
simple_form

Version:
2.0.1

ID:
CVE-2019-16676

LINK

simple_form Gem for Ruby Incorrect Access Control for forms based on user input

DESCRIPTION:

Simple Form before 5.0 has Incorrect Access Control in `file_method?` in `lib/simple_form/form_builder.rb`, because a user-supplied string is invoked as a method call.
This only happens for pages that build forms based on user input.

VULNERABLE GEM: SPROCKETS@2.2.3

Name:
sprockets

Version:
2.2.3

ID:
CVE-2018-3760

LINK

Path Traversal in Sprockets

DESCRIPTION:

Specially crafted requests can be used to access files that exist on the filesystem that is outside an application's root directory, when the Sprockets server is used in production.
All users running an affected release should either upgrade or use one of the work arounds immediately.
Workaround: In Rails applications, work around this issue, set `config.assets.compile = false` and `config.public_file_server.enabled = true` in an initializer and precompile the assets.
This work around will not be possible in all hosting environments and upgrading is advised.

VULNERABLE GEM: TZINFO@0.3.52

Name:
tzinfo

Version:
0.3.52

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`.

VULNERABLE GEM: UGLIFIER@1.2.4

Name:
uglifier

Version:
1.2.4

ID:
CVE-2015-8857

LINK

---

uglifier incorrectly handles non-boolean comparisons during minification

DESCRIPTION:

The upstream library for the Ruby uglifier gem, UglifyJS, is affected by a vulnerability that allows a specially crafted Javascript file to have altered functionality after minification.
This bug, found in UglifyJS versions 2.4.23 and earlier, was demonstrated to allow potentially malicious code to be hidden within secure code, and activated by the minification process.
For more information, consult: * https://zyan.scripts.mit.edu/blog/backdooring-js

CWE: 254 - 7PK - Security Features