FastRuby
Audit Tool

44 Vulnerabilities found on your file

Advisories

44

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: 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_ww
w_form_component(id.to_s)}#{format_extension}#{query_string(qu
ery_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: 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-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

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: 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-2015-1840

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: 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-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

SMTP command injection

DESCRIPTION:

Because Mail does not disallow CRLF in email addresses, an attacker can inject SMTP commands in specially crafted email addresses passed to RCPT TO and MAIL FROM.
Not affected by this vulnerability: * Ruby 2.4.0+ with a fix for CVE-2015-9096. * Applications that do not use SMTP delivery. * Applications that validate email addresses to not include CRLF.
The injection attack is described in Terada, Takeshi. "SMTP Injection via Recipient Email Addresses." 2015. The attacks described in the paper (Terada, p. 4) can be applied to the library without any modification.

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-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-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-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-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:
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:
CVE-2020-26247

LINK

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

DESCRIPTION:

DESCRIPTION

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

AFFECTED VERSIONS

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

MITIGATION

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

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

So if your previous code was:

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

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

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

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

VULNERABLE GEM: NOKOGIRI@1.5.5

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-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-2017-5029

LINK

Nokogiri gem contains two upstream vulnerabilities in libxslt 1.1.29

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-2013-6461

LINK

Nokogiri Gem for Ruby External Entity (XXE) Expansion Remote DoS

DESCRIPTION:

Nokogiri gem for Ruby contains an flaw that is triggered during the parsing of XML data. The issue is due to an incorrectly configured XML parser accepting XML external entities from an untrusted source. By sending

specially crafted XML data, a remote attacker can cause an infinite loop and crash the program.

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: NOKOGIRI@1.5.5

Name:

Version:

nokogiri **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: **Version:**
nokogiri **1.5.5**

ID:
CVE-2013-6460

LINK

Nokogiri Gem for JRuby Crafted XML Document Handling Infinite Loop Remote DoS

DESCRIPTION:

Nokogiri Gem for JRuby contains a flaw that may allow a remote denial of service. The issue is triggered when handling a specially crafted XML document, which can result in an infinite loop. This may allow a context-dependent attacker to crash the server.

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: PAPERCLIP@3.3.1

Name:
paperclip

Version:
3.3.1

ID:
OSVDB-103151

LINK

Paperclip Gem for Ruby contains a flaw

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

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-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-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-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-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-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:

Version:

rake **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: 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-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-2018-1000544

LINK

Directory Traversal in rubyzip

DESCRIPTION:

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: UGLIFIER@1.2.4

Name:
uglifier

Version:
1.2.4

ID:
OSVDB-126747

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/