Security

is hard

André Arko

@indirect

Security

is **hard**

but

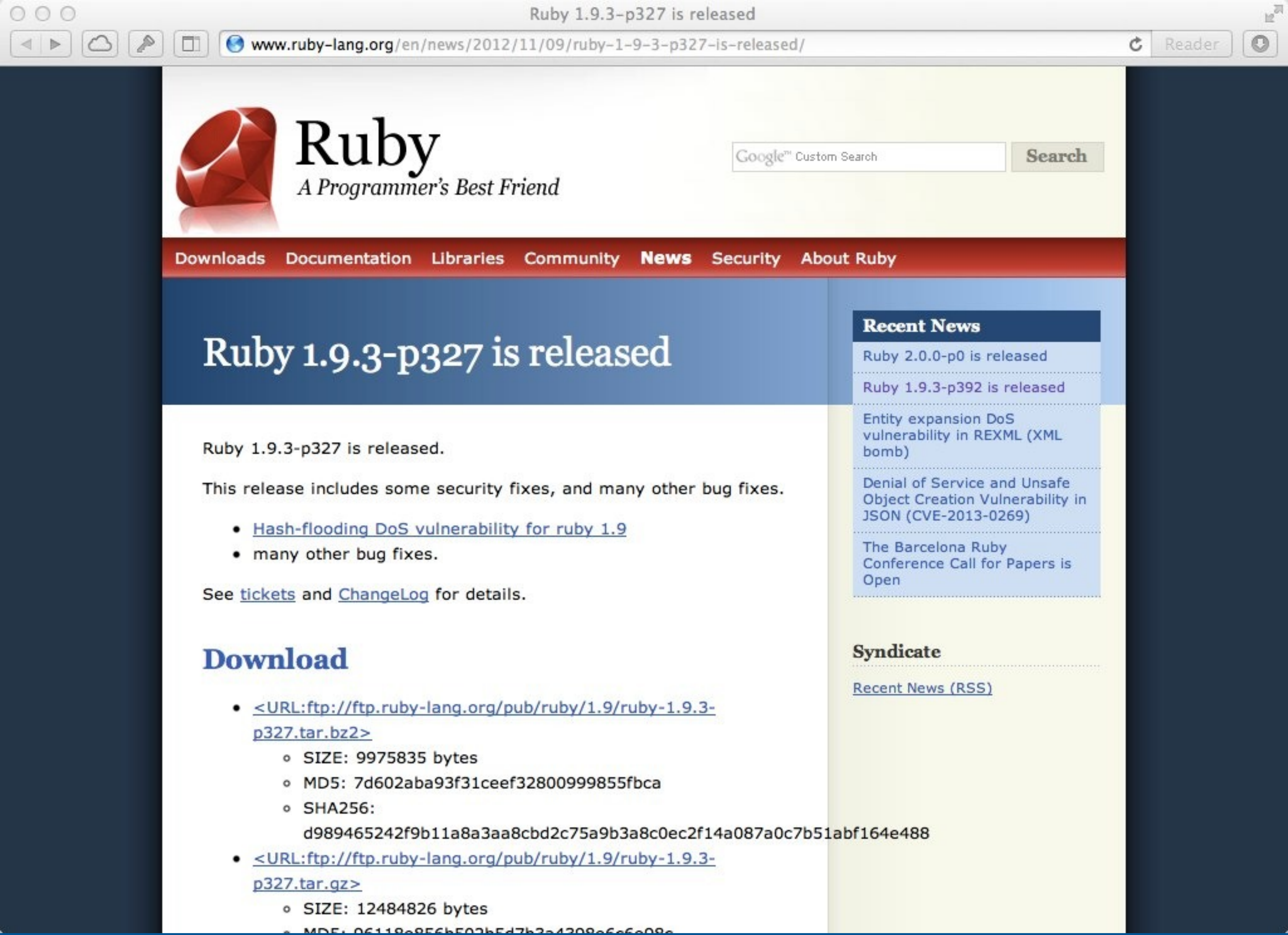
we can’t

go shopping

😢

Ruby

security releases


Ruby 1.9.3-p327 is released
www.ruby-lang.org/en/news/2012/11/09/ruby-1-9-3-p327-is-released/
Reader
Ruby
A Programmer's Best Friend
Google™ Custom Search
Search
Downloads
Documentation
Libraries
Community
News
Security
About Ruby

Ruby 1.9.3-p327 is released

Ruby 1.9.3-p327 is released.

This release includes some security fixes, and many other bug fixes.

- Hash-flooding DoS vulnerability for ruby 1.9
- many other bug fixes.

See tickets and ChangeLog for details.

Download

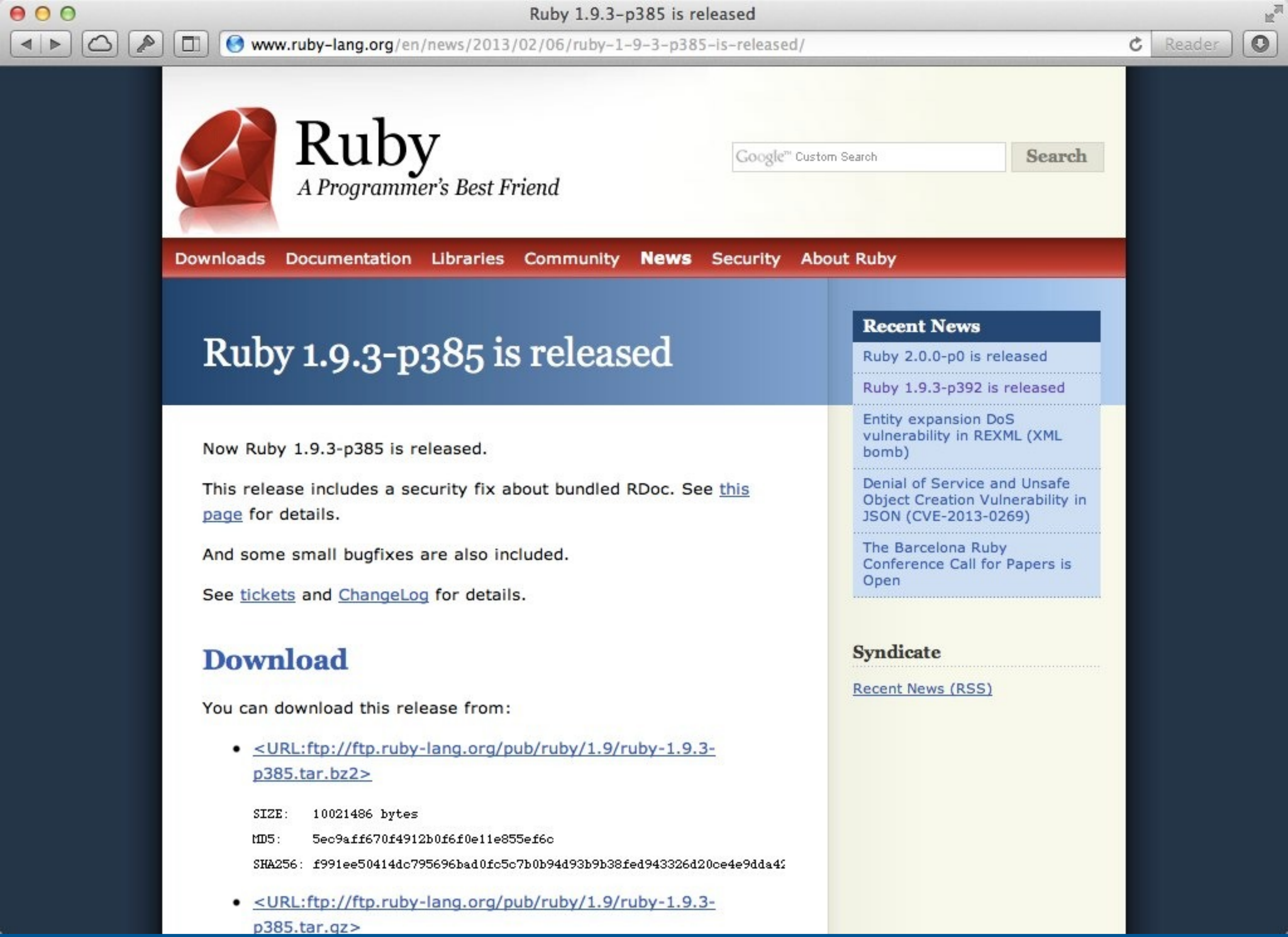
- <URL:ftp://ftp.ruby-lang.org/pub/ruby/1.9/ruby-1.9.3-p327.tar.bz2>
  - SIZE: 9975835 bytes
  - MD5: 7d602aba93f31ceef32800999855fbca
  - SHA256: d989465242f9b11a8a3aa8cbd2c75a9b3a8c0ec2f14a087a0c7b51abf164e488
- <URL:ftp://ftp.ruby-lang.org/pub/ruby/1.9/ruby-1.9.3-p327.tar.gz>
  - SIZE: 12484826 bytes

Recent News

- Ruby 2.0.0-p0 is released
- Ruby 1.9.3-p392 is released
- Entity expansion DoS vulnerability in REXML (XML bomb)
- Denial of Service and Unsafe Object Creation Vulnerability in JSON (CVE-2013-0269)
- The Barcelona Ruby Conference Call for Papers is Open

Syndicate

Recent News (RSS)


Ruby 1.9.3-p385 is released
www.ruby-lang.org/en/news/2013/02/06/ruby-1-9-3-p385-is-released/
Reader
Ruby
A Programmer's Best Friend
Google™ Custom Search
Search
Downloads
Documentation
Libraries
Community
News
Security
About Ruby

Ruby 1.9.3-p385 is released

Now Ruby 1.9.3-p385 is released.

This release includes a security fix about bundled RDoc. See this page for details.

And some small bugfixes are also included.

See tickets and ChangeLog for details.

Download

You can download this release from:

- <URL:ftp://ftp.ruby-lang.org/pub/ruby/1.9/ruby-1.9.3-p385.tar.bz2>

```
SIZE:   10021486 bytes
MD5:    5ec9aff670f4912b0f6f0e11e855ef6c
SHA256: f991ee50414dc795696bad0fc5c7b0b94d93b9b38fed943326d20ce4e9dda4
```

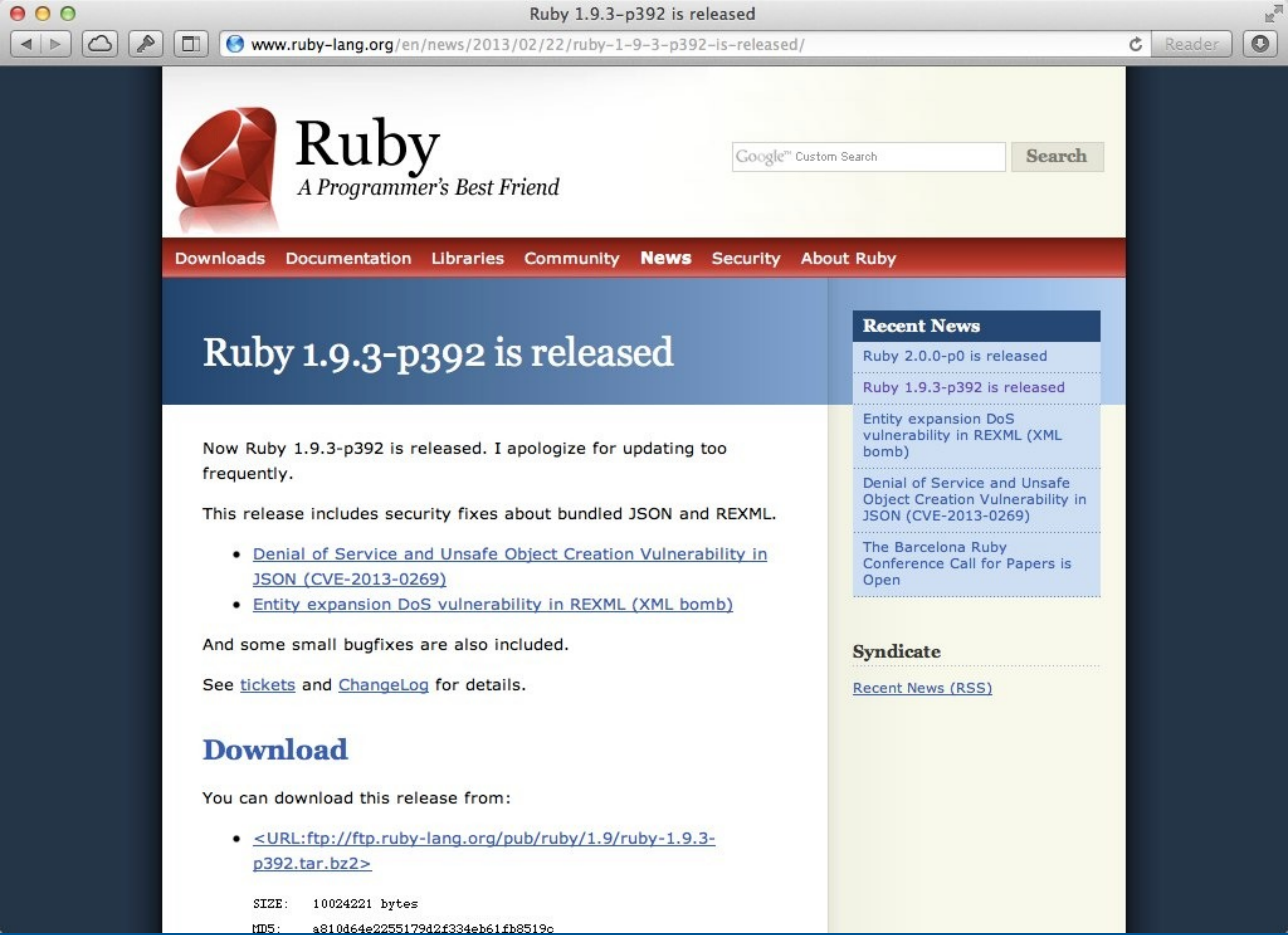
- <URL:ftp://ftp.ruby-lang.org/pub/ruby/1.9/ruby-1.9.3-p385.tar.gz>

Recent News

- Ruby 2.0.0-p0 is released
- Ruby 1.9.3-p392 is released
- Entity expansion DoS vulnerability in REXML (XML bomb)
- Denial of Service and Unsafe Object Creation Vulnerability in JSON (CVE-2013-0269)
- The Barcelona Ruby Conference Call for Papers is Open

Syndicate

Recent News (RSS)


Ruby 1.9.3-p392 is released
www.ruby-lang.org/en/news/2013/02/22/ruby-1-9-3-p392-is-released/
Reader
Ruby
A Programmer's Best Friend
Google™ Custom Search
Search
Downloads
Documentation
Libraries
Community
News
Security
About Ruby

Ruby 1.9.3-p392 is released

Now Ruby 1.9.3-p392 is released. I apologize for updating too frequently.

This release includes security fixes about bundled JSON and REXML.

- Denial of Service and Unsafe Object Creation Vulnerability in JSON (CVE-2013-0269)
- Entity expansion DoS vulnerability in REXML (XML bomb)

And some small bugfixes are also included.

See tickets and ChangeLog for details.

Download

You can download this release from:

- <URL:ftp://ftp.ruby-lang.org/pub/ruby/1.9/ruby-1.9.3-p392.tar.bz2>

```
SIZE:   10024221 bytes
MD5:    a810d64e2255179d2f334eb61fb8519c
```

Recent News

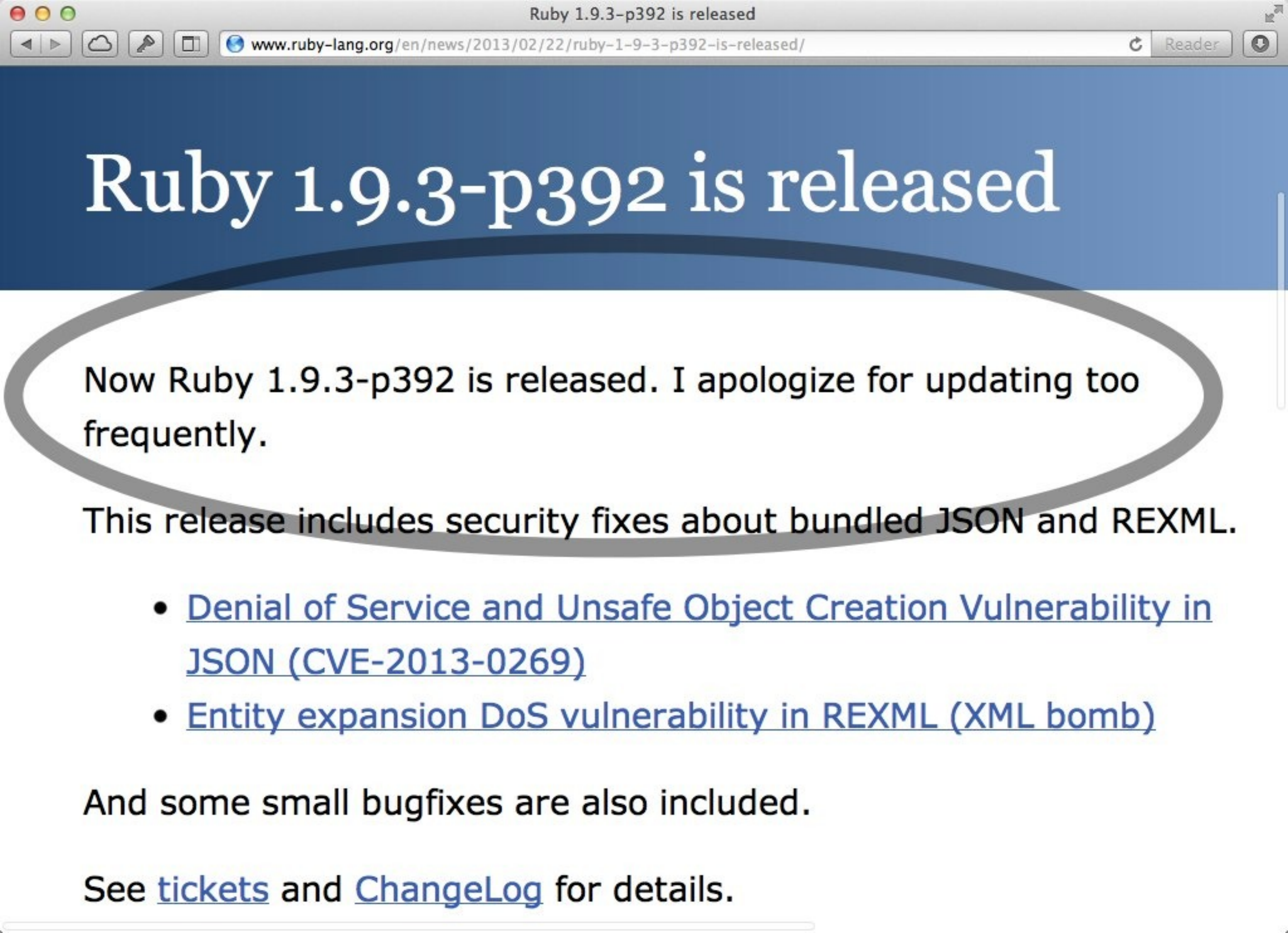
- Ruby 2.0.0-p0 is released
- Ruby 1.9.3-p392 is released
- Entity expansion DoS vulnerability in REXML (XML bomb)
- Denial of Service and Unsafe Object Creation Vulnerability in JSON (CVE-2013-0269)
- The Barcelona Ruby Conference Call for Papers is Open

Syndicate

Recent News (RSS)

this is not

normal


Ruby 1.9.3-p392 is released
www.ruby-lang.org/en/news/2013/02/22/ruby-1-9-3-p392-is-released/
Reader

Ruby 1.9.3-p392 is released

Now Ruby 1.9.3-p392 is released. I apologize for updating too frequently.

This release includes security fixes about bundled JSON and REXML.

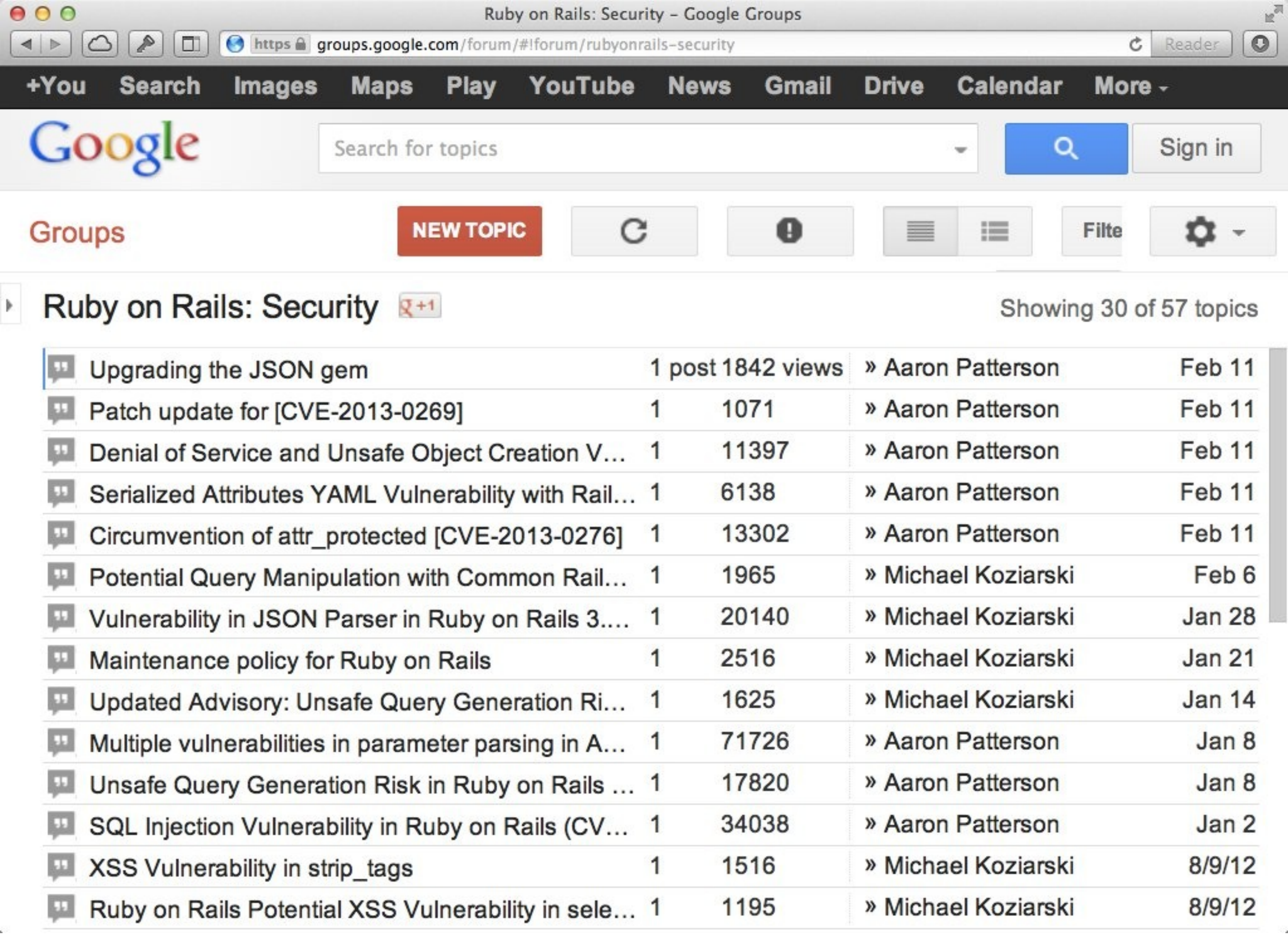
- Denial of Service and Unsafe Object Creation Vulnerability in JSON (CVE-2013-0269)
- Entity expansion DoS vulnerability in REXML (XML bomb)

And some small bugfixes are also included.

See tickets and ChangeLog for details.

Rails

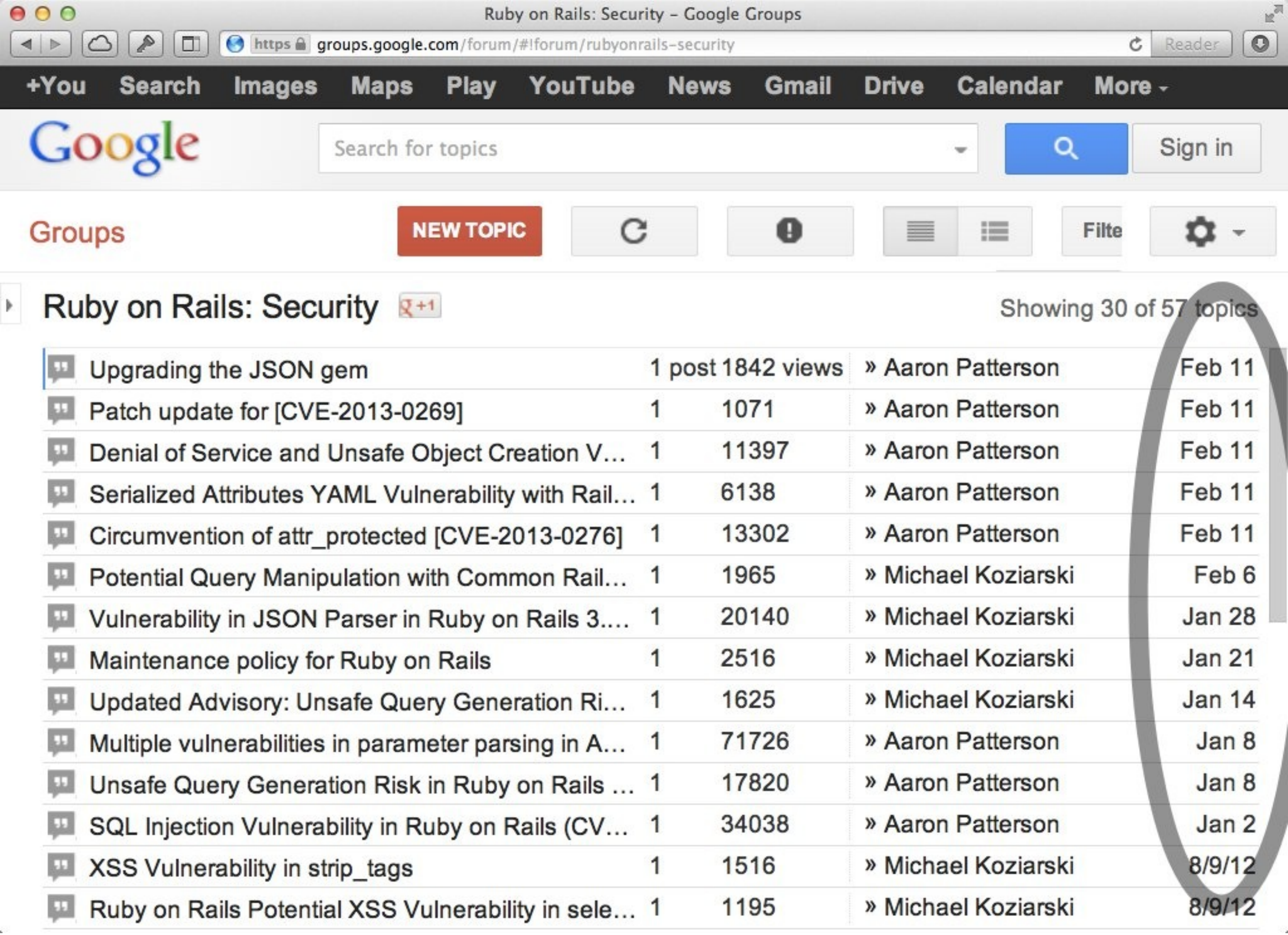
security releases



Ruby on Rails: Security

Showing 30 of 57 topics

Upgrading the JSON gem	1 post	1842 views	» Aaron Patterson	Feb 11
Patch update for [CVE-2013-0269]	1	1071	» Aaron Patterson	Feb 11
Denial of Service and Unsafe Object Creation V…	1	11397	» Aaron Patterson	Feb 11
Serialized Attributes YAML Vulnerability with Rail…	1	6138	» Aaron Patterson	Feb 11
Circumvention of attr_protected [CVE-2013-0276]	1	13302	» Aaron Patterson	Feb 11
Potential Query Manipulation with Common Rail…	1	1965	» Michael Koziarski	Feb 6
Vulnerability in JSON Parser in Ruby on Rails 3.…	1	20140	» Michael Koziarski	Jan 28
Maintenance policy for Ruby on Rails	1	2516	» Michael Koziarski	Jan 21
Updated Advisory: Unsafe Query Generation Ri…	1	1625	» Michael Koziarski	Jan 14
Multiple vulnerabilities in parameter parsing in A…	1	71726	» Aaron Patterson	Jan 8
Unsafe Query Generation Risk in Ruby on Rails …	1	17820	» Aaron Patterson	Jan 8
SQL Injection Vulnerability in Ruby on Rails (CV…	1	34038	» Aaron Patterson	Jan 2
XSS Vulnerability in strip_tags	1	1516	» Michael Koziarski	8/9/12
Ruby on Rails Potential XSS Vulnerability in sele…	1	1195	» Michael Koziarski	8/9/12



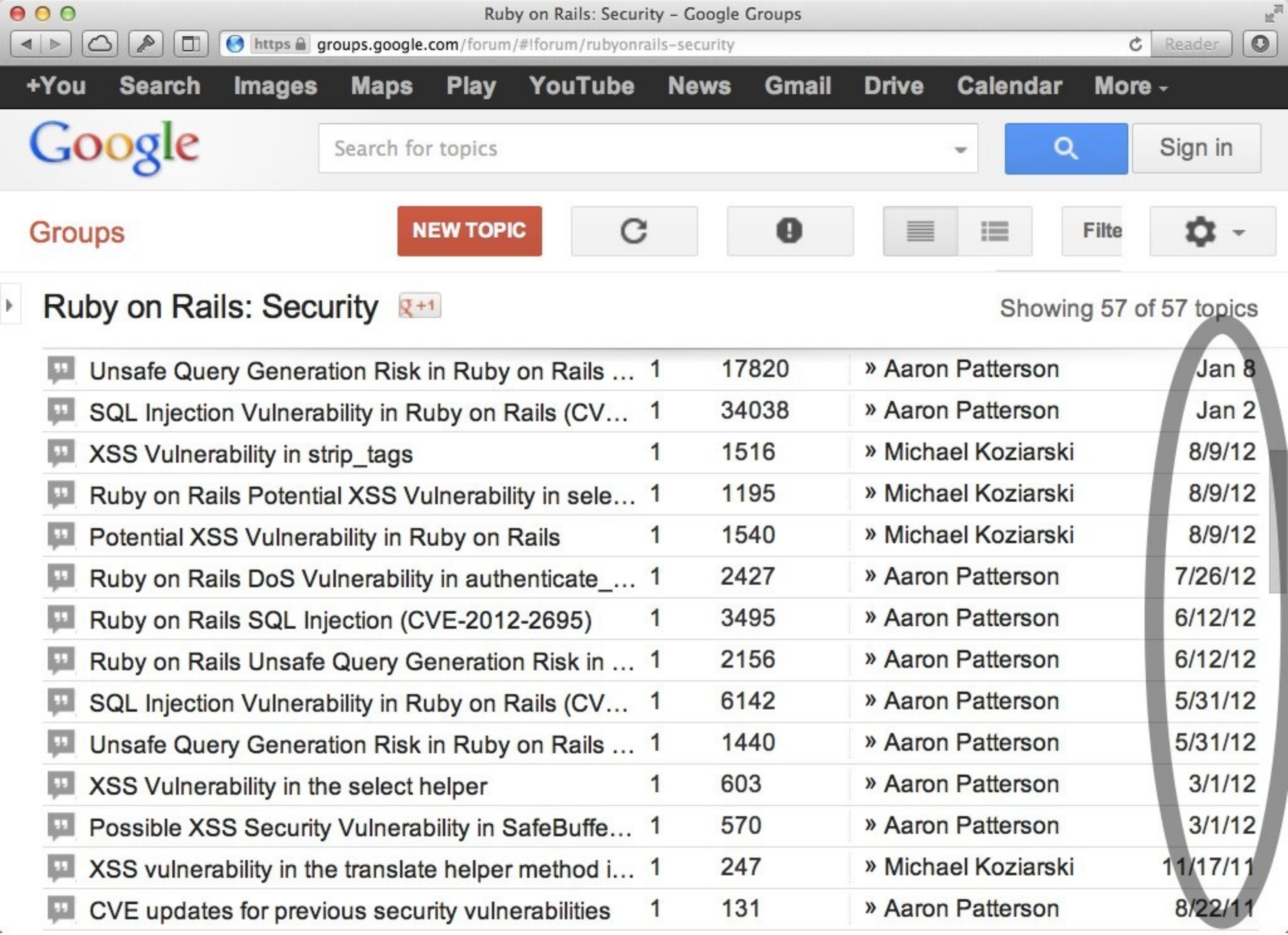
Ruby on Rails: Security

Showing 30 of 57 topics

Upgrading the JSON gem	1 post	1842 views	» Aaron Patterson	Feb 11
Patch update for [CVE-2013-0269]	1	1071	» Aaron Patterson	Feb 11
Denial of Service and Unsafe Object Creation V…	1	11397	» Aaron Patterson	Feb 11
Serialized Attributes YAML Vulnerability with Rail…	1	6138	» Aaron Patterson	Feb 11
Circumvention of attr_protected [CVE-2013-0276]	1	13302	» Aaron Patterson	Feb 11
Potential Query Manipulation with Common Rail…	1	1965	» Michael Koziarski	Feb 6
Vulnerability in JSON Parser in Ruby on Rails 3.…	1	20140	» Michael Koziarski	Jan 28
Maintenance policy for Ruby on Rails	1	2516	» Michael Koziarski	Jan 21
Updated Advisory: Unsafe Query Generation Ri…	1	1625	» Michael Koziarski	Jan 14
Multiple vulnerabilities in parameter parsing in A…	1	71726	» Aaron Patterson	Jan 8
Unsafe Query Generation Risk in Ruby on Rails …	1	17820	» Aaron Patterson	Jan 8
SQL Injection Vulnerability in Ruby on Rails (CV…	1	34038	» Aaron Patterson	Jan 2
XSS Vulnerability in strip_tags	1	1516	» Michael Koziarski	8/9/12
Ruby on Rails Potential XSS Vulnerability in sele…	1	1195	» Michael Koziarski	8/9/12

this isn’t

normal either



Ruby on Rails: Security

Showing 57 of 57 topics

Unsafe Query Generation Risk in Ruby on Rails …	1	17820	» Aaron Patterson	Jan 8
SQL Injection Vulnerability in Ruby on Rails (CV…	1	34038	» Aaron Patterson	Jan 2
XSS Vulnerability in strip_tags	1	1516	» Michael Koziarski	8/9/12
Ruby on Rails Potential XSS Vulnerability in sele…	1	1195	» Michael Koziarski	8/9/12
Potential XSS Vulnerability in Ruby on Rails	1	1540	» Michael Koziarski	8/9/12
Ruby on Rails DoS Vulnerability in authenticate_…	1	2427	» Aaron Patterson	7/26/12
Ruby on Rails SQL Injection (CVE-2012-2695)	1	3495	» Aaron Patterson	6/12/12
Ruby on Rails Unsafe Query Generation Risk in …	1	2156	» Aaron Patterson	6/12/12
SQL Injection Vulnerability in Ruby on Rails (CV…	1	6142	» Aaron Patterson	5/31/12
Unsafe Query Generation Risk in Ruby on Rails …	1	1440	» Aaron Patterson	5/31/12
XSS Vulnerability in the select helper	1	603	» Aaron Patterson	3/1/12
Possible XSS Security Vulnerability in SafeBuffe…	1	570	» Aaron Patterson	3/1/12
XSS vulnerability in the translate helper method i…	1	247	» Michael Koziarski	11/17/11
CVE updates for previous security vulnerabilities	1	131	» Aaron Patterson	8/22/11

wait

what’s a CVE?

common vulnerabilities and exposures

numbering authorities

apple

adobe

cisco

redhat

etc.

cve.mitre.org

nvd.nist.gov

minaswan

security?

vulnerabilities?

dhh + rails

not as nice

dhh + rails

but we can learn from them

so many

gems

for everything

so many

chances for

security issues

rubygems

bundler

json

rexml

rack

arel

activerecord

actionpack

activesupport

rdoc (rdoc?! yup.)

what

should we do?

updating

is a pain

updating

blocks feature

development

updating

is insurance

a small cost

to mitigate risk

without it

failures are

catastrophic

disclosure

liability

lawyers

updating

is hard work

😣

but

updating is worth it

update

sleep well at night

😴

reporting

security issues

responsible

disclosure

the worst

except for all the other options

the best yet

because everyone
ends up *unhappy*

but

no one ends

up **screwed**

disclosure

companies hate it

responsible

clever, triumphant

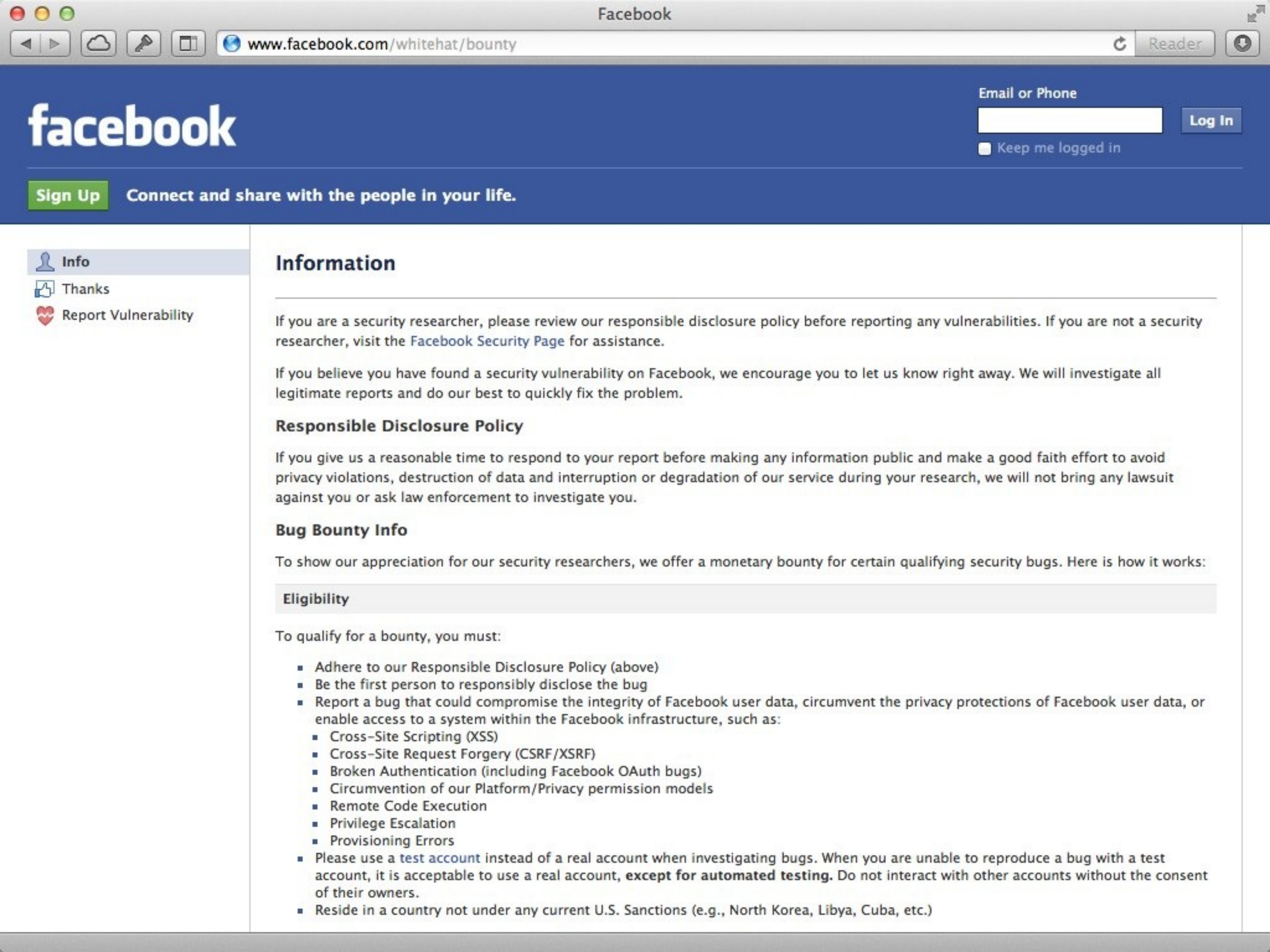
hackers hate it

rewards! 💰

rewards! 💰

maybe everyone ends up happy?

facebook



Information

If you are a security researcher, please review our responsible disclosure policy before reporting any vulnerabilities. If you are not a security researcher, visit the Facebook Security Page for assistance.

If you believe you have found a security vulnerability on Facebook, we encourage you to let us know right away. We will investigate all legitimate reports and do our best to quickly fix the problem.

Responsible Disclosure Policy

If you give us a reasonable time to respond to your report before making any information public and make a good faith effort to avoid privacy violations, destruction of data and interruption or degradation of our service during your research, we will not bring any lawsuit against you or ask law enforcement to investigate you.

Bug Bounty Info

To show our appreciation for our security researchers, we offer a monetary bounty for certain qualifying security bugs. Here is how it works:

Eligibility

To qualify for a bounty, you must:

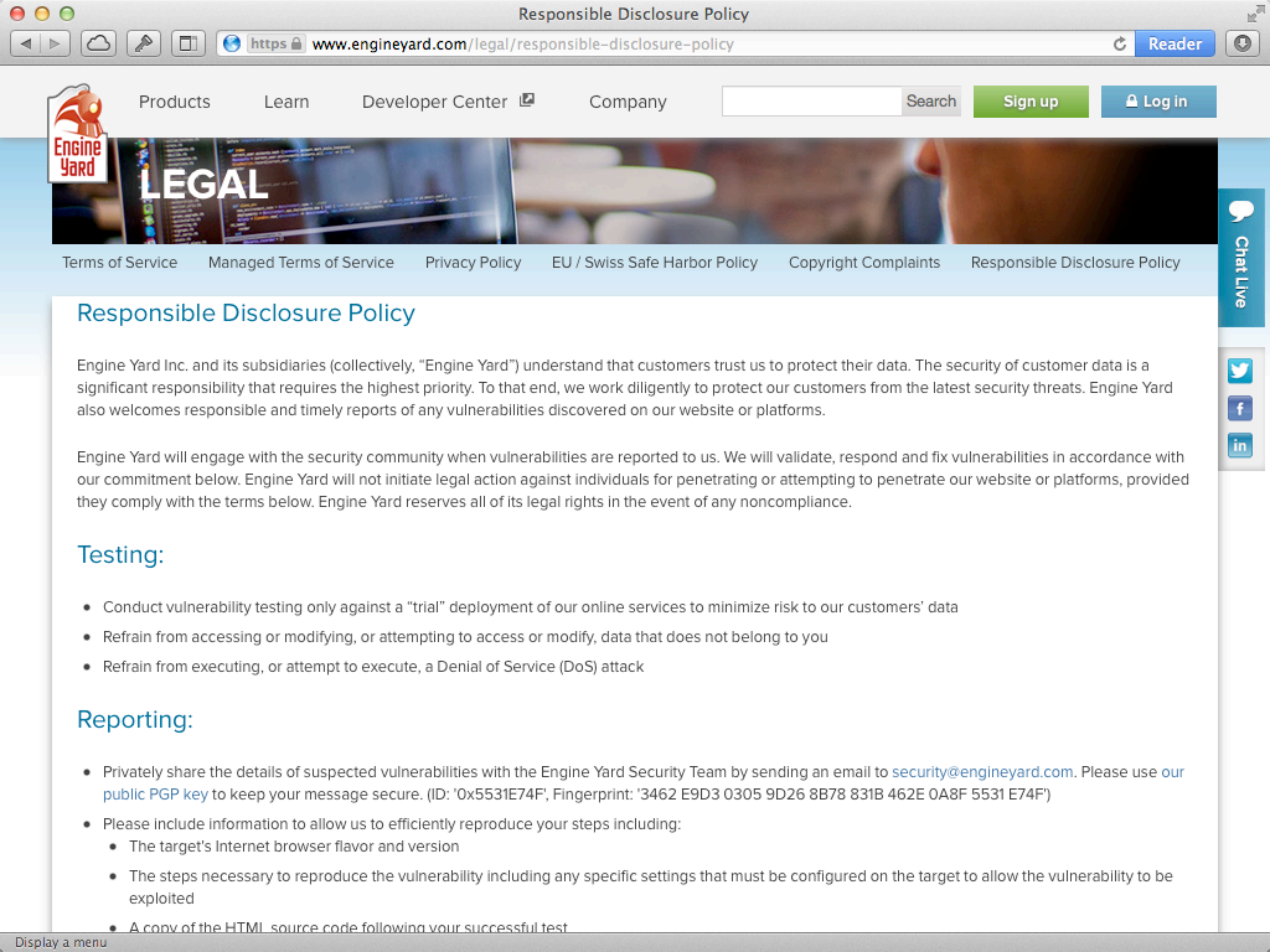
- Adhere to our Responsible Disclosure Policy (above)
- Be the first person to responsibly disclose the bug
- Report a bug that could compromise the integrity of Facebook user data, circumvent the privacy protections of Facebook user data, or enable access to a system within the Facebook infrastructure, such as:
  - Cross-Site Scripting (XSS)
  - Cross-Site Request Forgery (CSRF/XSRF)
  - Broken Authentication (including Facebook OAuth bugs)
  - Circumvention of our Platform/Privacy permission models
  - Remote Code Execution
  - Privilege Escalation
  - Provisioning Errors
- Please use a test account instead of a real account when investigating bugs. When you are unable to reproduce a bug with a test account, it is acceptable to use a real account, **except for automated testing.** Do not interact with other accounts without the consent of their owners.
- Reside in a country not under any current U.S. Sanctions (e.g., North Korea, Libya, Cuba, etc.)

facebook

$500 minimum

no maximum

engine yard



Responsible Disclosure Policy

Engine Yard Inc. and its subsidiaries (collectively, "Engine Yard") understand that customers trust us to protect their data. The security of customer data is a significant responsibility that requires the highest priority. To that end, we work diligently to protect our customers from the latest security threats. Engine Yard also welcomes responsible and timely reports of any vulnerabilities discovered on our website or platforms.

Engine Yard will engage with the security community when vulnerabilities are reported to us. We will validate, respond and fix vulnerabilities in accordance with our commitment below. Engine Yard will not initiate legal action against individuals for penetrating or attempting to penetrate our website or platforms, provided they comply with the terms below. Engine Yard reserves all of its legal rights in the event of any noncompliance.

Testing:

- Conduct vulnerability testing only against a "trial" deployment of our online services to minimize risk to our customers' data
- Refrain from accessing or modifying, or attempting to access or modify, data that does not belong to you
- Refrain from executing, or attempt to execute, a Denial of Service (DoS) attack

Reporting:

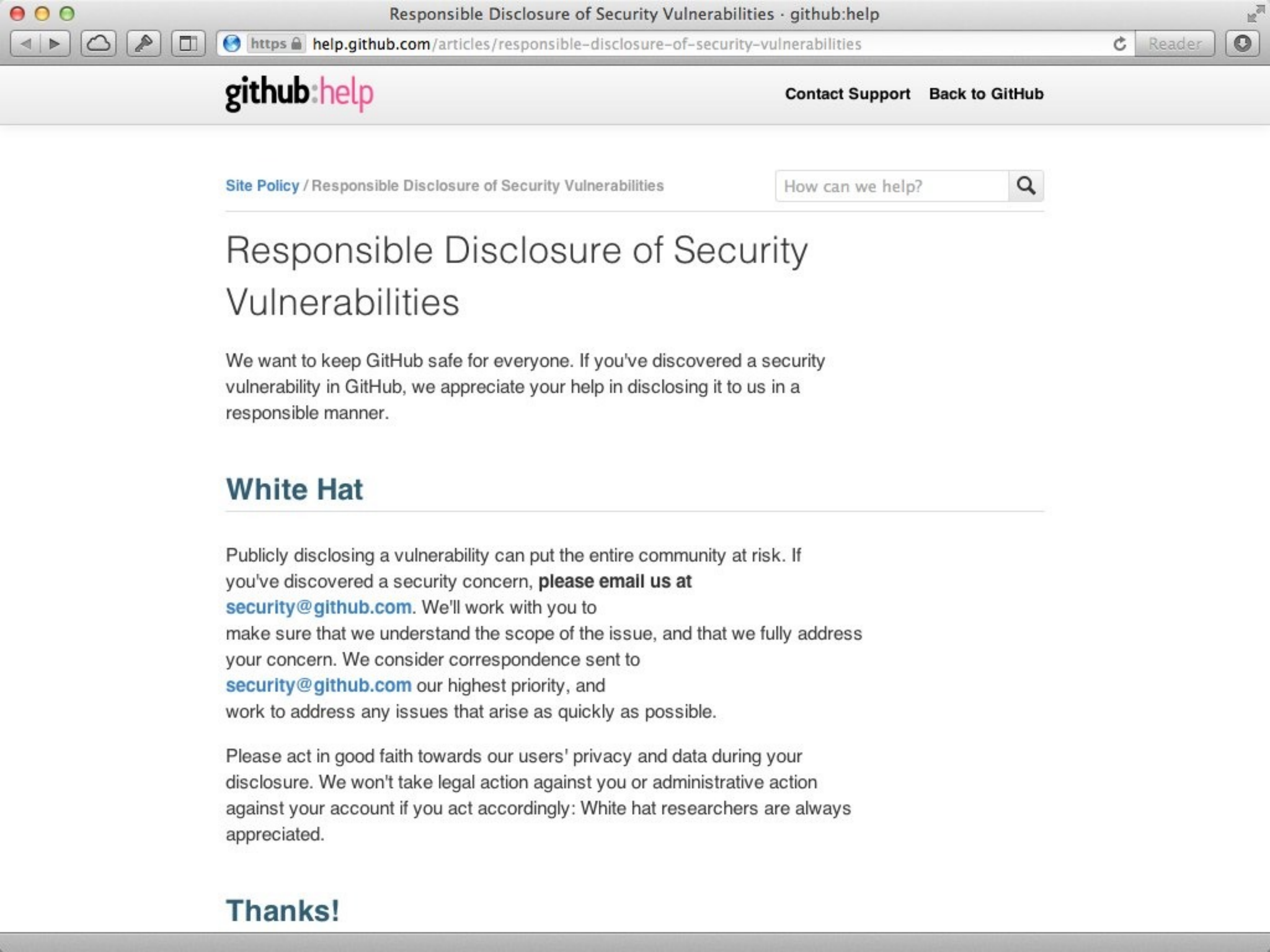
- Privately share the details of suspected vulnerabilities with the Engine Yard Security Team by sending an email to security@engineyard.com. Please use our public PGP key to keep your message secure. (ID: '0x5531E74F', Fingerprint: '3462 E9D3 0305 9D26 8B78 831B 462E 0A8F 5531 E74F')
- Please include information to allow us to efficiently reproduce your steps including:
  - The target's Internet browser flavor and version
  - The steps necessary to reproduce the vulnerability including any specific settings that must be configured on the target to allow the vulnerability to be exploited
  - A copy of the HTML source code following your successful test

engine yard

no compensation

$0 maximum

github


Responsible Disclosure of Security Vulnerabilities · github:help
https
help.github.com/articles/responsible-disclosure-of-security-vulnerabilities
Reader
github:help
Contact Support
Back to GitHub
How can we help?

Site Policy / Responsible Disclosure of Security Vulnerabilities

Responsible Disclosure of Security Vulnerabilities

We want to keep GitHub safe for everyone. If you've discovered a security vulnerability in GitHub, we appreciate your help in disclosing it to us in a responsible manner.

White Hat

Publicly disclosing a vulnerability can put the entire community at risk. If you've discovered a security concern, **please email us at security@github.com**. We'll work with you to make sure that we understand the scope of the issue, and that we fully address your concern. We consider correspondence sent to security@github.com our highest priority, and work to address any issues that arise as quickly as possible.

Please act in good faith towards our users' privacy and data during your disclosure. We won't take legal action against you or administrative action against your account if you act accordingly: White hat researchers are always appreciated.

Thanks!

github

no stated policy

$? maximum

anyway, back to

you

what if you

find a bug?

ask yourself two questions

can I access
something
I shouldn’t?

can I **disable**
something for
other people?

if the answer was **yes**

disclose responsibly

contact an author before reporting publicly

look for

a security policy

email in gemspec

email on github

work together

have empathy

if all else fails

if all else fails

fix it!

finally,

what about your gems?

your gems

are security vulnerabilities waiting to happen

unless

your code is perfect

(and you want to buy this real estate in Florida)

easy

sympathetic discoverer

easy

write fix, review fix

release + announce

medium

problem in the wild

medium

announce if safe

fix ASAP, test fix

release + announce

hard

researcher out for glory

hard

respond ASAP

set expectations

update every **24-48h**

fix + release + thanks

make it

as easy as possible

personally

gemspec email

github email

on a team

security address

PGP key

disclosure policy

ecosystem

mailing list for announcing security issues and releases bit.ly/ruby-sec-ann

we can

go shopping

🎩👠👞👗

👖👔👕🍰

questions?