

Hacking Rails

@jimmynguyc

Why?



<http://phrack.org/>

::: Attacking Ruby on Rails Applications :::

Papers:

Strauss - The Fall of Hacker Groups (2014-04-04)

fGI - Revisiting Mac OS X Kernel Rootkits (2014-04-18)

aaron portnoy - Adobe Shockwave - A case study on memory disclosure (2014-05-28)

nemo - Modern Objective-C Exploitation Techniques (2015-04-17)

Alisa Esage - Self-patching Microsoft XML with misalignments and factorials (2015-08-26)

kernnel - Internet Voting: A Requiem for the Dream (2015-09-09)

joernchen - Attacking Ruby on Rails Applications (2015-10-20)

Title : Attacking Ruby on Rails Applications

Author : joernchen

Date : October 20, 2015

```
|=====|
|-----=[ Attacking Ruby on Rails Applications ]-----|
|=====|
|-----=[ joernchen of Phenoelit ]-----|
|-----=[ joernchen@phenoelit.de ]-----|
|=====|
```

--[Table of contents

0 - Intro

1 - A Brief Overview

1.1 - User input

1.1.1 - POST/PUT/GET application/x-www-form-urlencoded

1.1.2 - Multiparameter attributes

1.1.3 - POST/PUT text/xml

1.1.4 - POST/PUT application/json

1.1.5 - GET vs. POST/PUT

2 - Common pitfalls

2.1 - Sessions

Ruby on Rails


- MVC based web application framework
- Convention over Configuration (CoC)
- Easy to learn
- Magical 🧙

Ruby on Rails

- Created by David Heinemeier Hansson (DHH)
- Extracted from Basecamp
- First release in 2004

Ruby on Rails


GitHub This repository Search Explore Features Enterprise Pricing Sign up Sign in

 rails / rails Watch 2,077 Star 28,559 Fork 11,550

Ruby on Rails <http://rubyonrails.org>

54,389 commits 37 branches 276 releases 2,888 contributors

Branch: master rails / +

 fxn Merge pull request #22443 from Wasserschlange/added_dollarsign_to_readme Latest commit 127a87c 10 hours ago

actionmailer	Move all nodoc methods to the private section	5 days ago
actionpack	Merge pull request #22371 from yui-knk/better_mount_error	19 hours ago
actionview	Merge pull request #21241 from pdg137/master	3 days ago

<> Code

Issues 404

Pull requests 466

Pulse

Graphs

HTTPS clone URL

@ 29-11-2015

Ruby on Rails

Top in Frameworks · Week beginning Nov 30th 2015

Name	10k	100k	Million	Entire Web	▲
PHP	↑4,613	↑44,534	↓587,001	↓41,700,644	
ASP.NET	↑2,627	↑25,781	↑310,025	↓38,210,545	
Shockwave Flash Embed	↑730	↑6,409	↓158,324	↓5,378,172	
Classic ASP	↑353	↑4,274	↑29,710	↓2,892,154	
Adobe Dreamweaver	↑433	↓4,880	↓84,983	↓2,771,343	
J2EE	↑1,384	↑7,413	↑77,944	↓1,982,820	
DAV	↑170	↑1,860	↓22,286	↓1,031,075	
ASP.NET MVC	↑671	↑3,923	↑24,060	↓905,838	
ASP.NET Ajax	↑928	↑8,850	↑80,867	↓895,791	
Ruby on Rails	↑709	↑3,956	↓34,075	↑860,644	
Perl	↑120	↑875	↓9,725	↓835,096	
Ruby on Rails Token	↑809	↑3,693	↓25,753	↓377,541	
Google PageSpeed Module	↑139	↑1,212	↓7,992	– 259,569	

#7

1,238,205

#10



source: buildwith.com

Ruby on Rails

More Usage = Bigger Target for Hackers

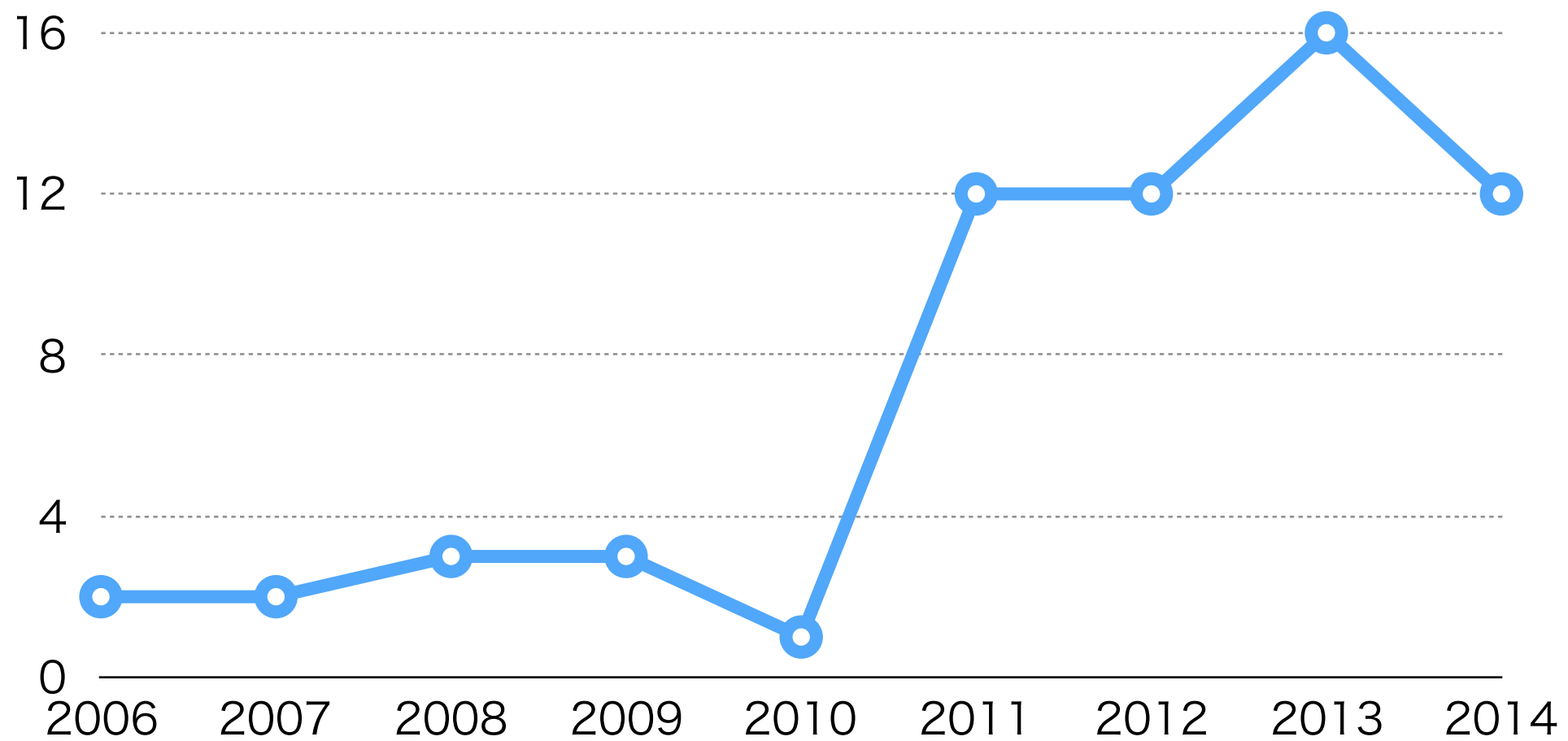


TARGET

CVE

- Dictionary of Common Vulnerabilities and Exposures (CVEs)

of CVEs



source: cvedetails.com

CVSS Scores

- Common Vulnerability Scoring System
- 0 to 10
- Based on 3 metric groups (base, temporal & environmental)
- Describes severity

CVSS Scores

Distribution of all vulnerabilities by CVSS Scores

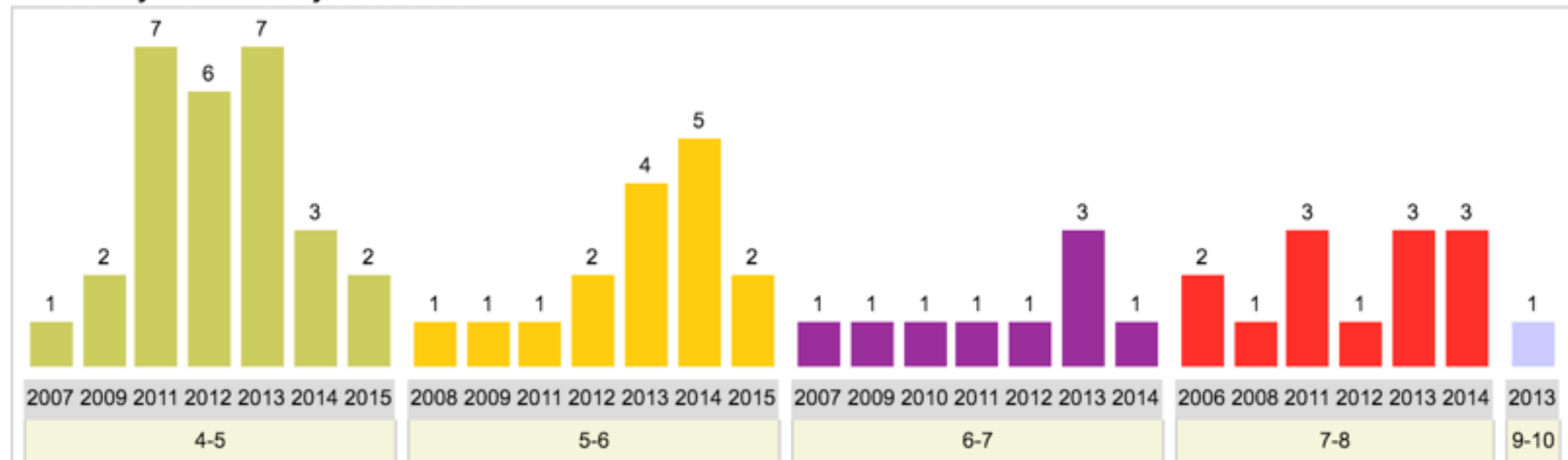
CVSS Score	Number Of Vulnerabilities	Percentage
0-1		0.00
1-2		0.00
2-3		0.00
3-4		0.00
4-5	28	41.80
5-6	16	23.90
6-7	9	13.40
7-8	13	19.40
8-9		0.00
9-10	1	1.50
Total	67	

Weighted Average CVSS Score: **6.2**

source: cvedetails.com

CVSS Scores

Vulnerability Distribution By CVSS Scores



source: cvedetails.com

IT'S THE END OF THE WORLD!!!



RUN FOR YOUR LIVES!!!

I.CAN.HAVE.CHEEZBURGER.COM 🍷 🍷 🍷

Responsible Disclosure



Responsible Disclosure

- All stakeholders agrees to allow a period of time for the vulnerability to be patched before publishing the details
- E.g. “Ruby on Rails: Security” [Google Group](#)

UPDATE RAILS ASAP ???



WTF DID I MISS ??!!

Let's Learn Something
Instead

Disclaimer

I am **not** responsible for any shenanigans that you might partake in as a direct or indirect result of the information that I'm about to tell you.

Approach

- Understand the exploit
- Look at the codes responsible
- Demo
- Understand the patch



Penetrating Testing Software

CVE-2013-0333

- CVSS 7.5
- Execute Code / Sql Injection / Bypass a restriction or similar
- Allows YAML to be loaded via JSON payload
- Basically exploiting `YAML::load(yaml)`

Code **R**esponsible

- `lib/action_dispatch/middleware/params_parser.rb`
- `lib/active_support/json/decoding.rb`
- `lib/active_support/json/backends/yaml.rb`

Demo

Patch

- <https://groups.google.com/forum/#!searchin/rubyonrails-security/2013-0333/rubyonrails-security/1h2DR63ViGo/GOUVaFeaF1IJ>

CVE-2013-0156

- CVSS 7.5
- Denial Of Service / Execute Code
- Allows YAML to be loaded via XML payload
- Exploiting `YAML::load(yaml)` ... again

Code **R**esponsible

- `lib/action_dispatch/middleware/params_parser.rb`
- `active_support/core_ext/hash/conversions.rb`
- `active_support/xml_mini.rb`

Demo

Patch

- <https://groups.google.com/forum/#!searchin/rubyonrails-security/CVE-2013-0156/rubyonrails-security/61bkgvnSGTQ/nehwjA8tQ8EJ>

DeviseDoor

joernchen/DeviseDoor

What Now?

- `gem install brakeman`
- Code Climate
- Read & subscribe to security blogs
- Good coding practices

Q&A

Thank you :)