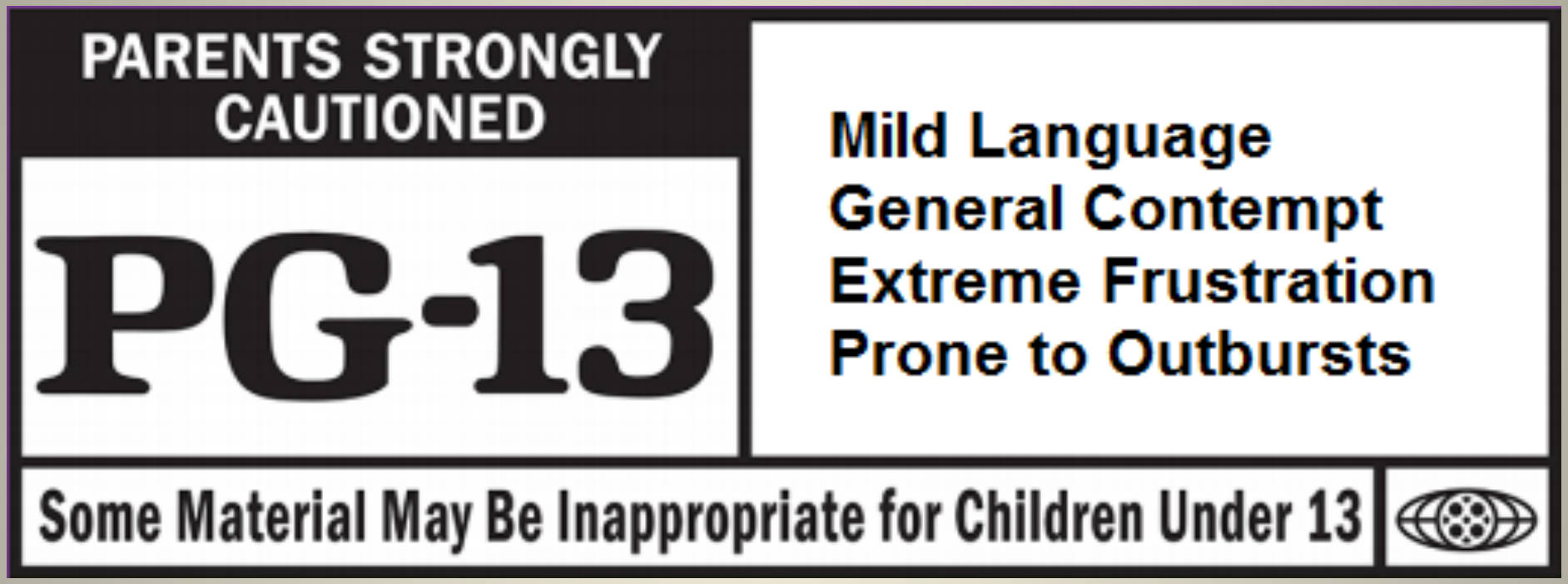
Buying Into the Bias: Why Vulnerability Statistics Suck

Steve Christey (MITRE) & Brian Martin (OSF)



(We speak for ourselves, not our employers)

Steve

Principal INFOSEC Engineer at MITRE

- CVE List Editor
- CWE Technical Lead
- Helped popularize ~~Responsible~~ Coordinated Disclosure

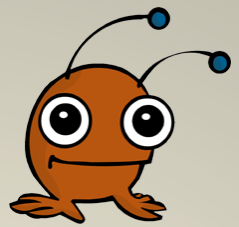
Random Facts

- Likes sushi. A lot.
- Annoys Bayesians and metrics geeks
- Is comfortable with 80% solutions
- Wants software-assurance "food labels"
- Favorite OSVDB ID: 79400

Things I've been doing

- Working on CVE-10K bug/feature
- Helping to build and empower the CVE content team for real longevity
- Trying to keep up with new vuln types
- Inching towards a "Grand Unified Theory" of vulnerabilities (i.e. tilting at windmills)
- Fighting the Terminological Cold War

Brian



President / COO of Open Security Foundation

- Content Manager for OSVDB
- President / COO of Open Security Foundation
- Director of Non-profit Activity at Risk Based Security

Random Facts

- First VDB maintained in 1994
- Joined OSVDB as volunteer in 2003
- CVE Editorial Board Member since 2008
- Has rescued 9 guinea pigs from shelters
- Favorite CVE ID: 2003-1599

Things I've been doing

- Vulnerability Databases
  - Everything about them.
  - Really, everything remotely related.
- History of vulnerabilities
- Vulnerability Disclosure Errata
- Bugs (of the software variety)

Challenge!

- Because overcoming 15 years of bad vulnerability stats wasn't enough…
- BlackHat is so competitive… and yet sometimes important topics are boring on screen…

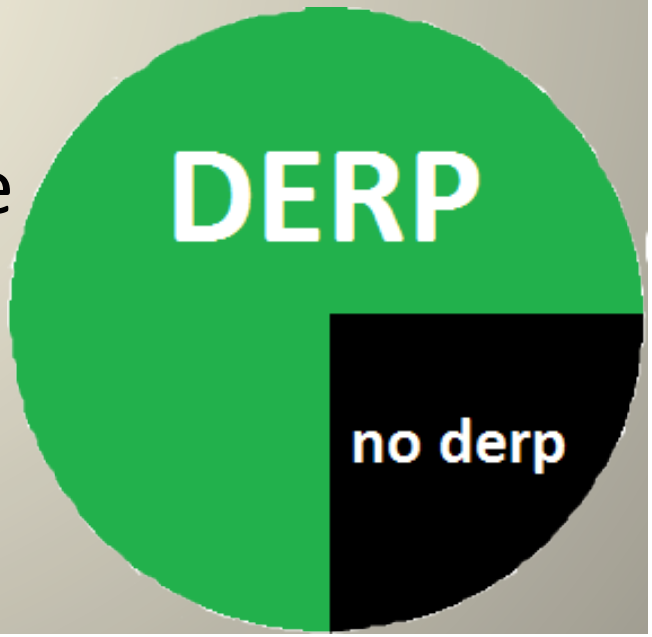
- We took your requests… <u>All 24 of them</u>.

Why do vuln stats matter?

- Favorite talking point for media whores
- Are used to make faulty comparisons about "security" (services, products, vendors)
- Security industry is about integrity. If our stats have none, where are we?
- How can we <u>really</u> tell if we're making progress?
- At least people don't make security decisions or shape their world view based on vulnerability statistics! *sob* *drink* *curse*

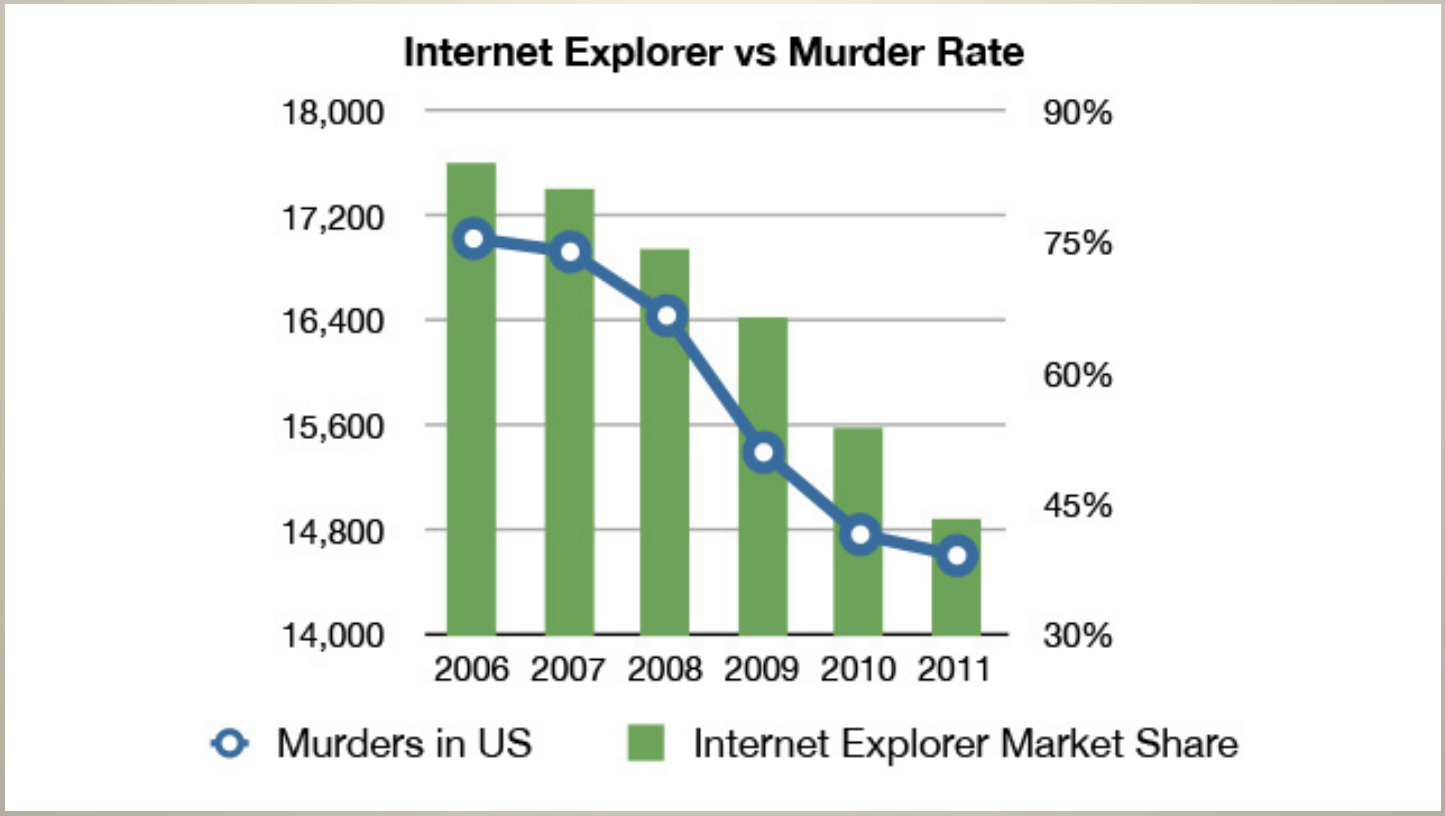
Why Vuln Total Stats are Worthless

- Inconsistent abstraction
- Significant gaps in coverage of vulns
- Specific focus and not caring about historical
- Bad stat analysis, no method for us to validate
- Sweeping assumptions about outside influences on stats or patterns
- Entries not created on root-cause



Why Vulnerability Stats Suck

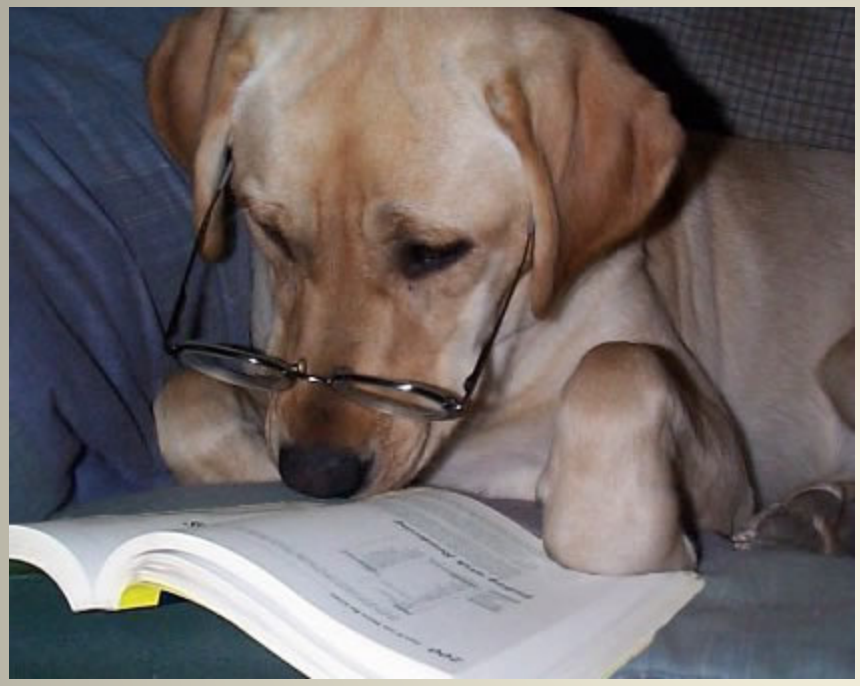
- Stats are presented without understanding the limits of the data
- Even if explanations are provided, correlation is confused with causation:



Talking Points

- Defining Bias
- Researcher Bias
- Vendor Bias
- VDB Bias
- Bad Stats
- Good(ish) Stats
- Conclusion

Can We Learn from Others?



- More mature fields have wrestled with bias
- Experimental design dates back to the 1700's

- Epidemiology: the study "of the patterns, causes, and effects of health and disease conditions," typically in human populations
  - Vulnerabilities are kind of like diseases?
  - Modern epidemiology dates back to 1854

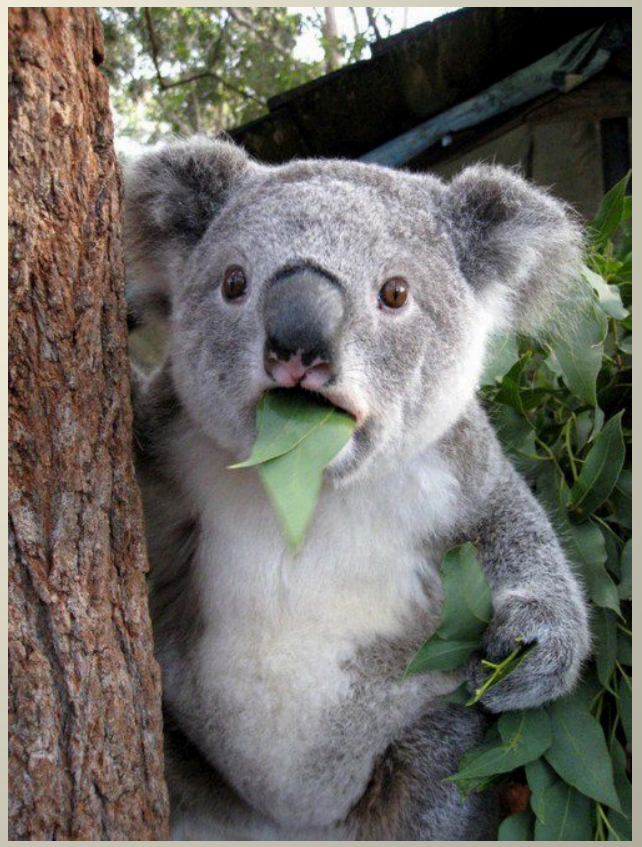
Disease Research: Epidemiology vs. Vulnerability Research

	Epidemiology	Vulnerability Research
Goal	Improve the public health	SAVE ALL THE THINGZ ON THA INTERWEBZ! * (attention whoring)
Objects of Study	People/Diseases	Software/Vulnerabilities
Populations	Groups of people	Groups of vulnerabilities (as seen in multi-vuln disclosures)
Measurement Devices (Tools of the Trade)	Blood pressure monitors, thermometers, lab tests, observation	Automated code scanners w/high FP/ FN rates, fuzzers, coffee-fueled malcontents staring at code at 3 AM
Publication Requirements	Refereed journals with peer review	Ability to send email
Sampling Methods	Using industry established methodologies and formal documentation.	Using wildly erratic methodologies, no standards for documentation or disclosure

* Goal not shared by all researchers. Please to be rolling with this, kthxbye

The Shocking Claim

- Bias and statistics in vulnerability research are far worse than it is in other disciplines
- At least people don't die (yet?), but still use vulnerable equipment:
  - SCADA
  - Airplanes
  - Automobiles
  - Medical Devices
  - Oh my…




Bias: You Have It

Bias: An Overview

- In statistics, “bias” effectively reflects the degree to which a statistic does not properly represent the entire population being measured
  - If there’s a lot of bias, then the measurement is highly suspect
- Many, many types and subtypes of bias
- Different fields have different terms

Four Main Types of Bias

- Selection Bias: what gets selected for study
- Publication Bias: what gets published (or not)
- Abstraction Bias: a term we ~~made up~~ *crafted* for how vulnerabilities are counted
  - Many fields count by "person" or other discrete physical objects. We can't unfortunately.
- Measurement Bias: introduced by inaccurate or imprecise "measurements"

Measure us biyatch!

Selection Bias

Selection Bias

- "there is an error in choosing the individuals or groups to take part in a scientific study [which leads to] distortion of a statistical analysis, resulting from the method of collecting samples. If the selection bias is not taken into account then certain conclusions drawn may be wrong."

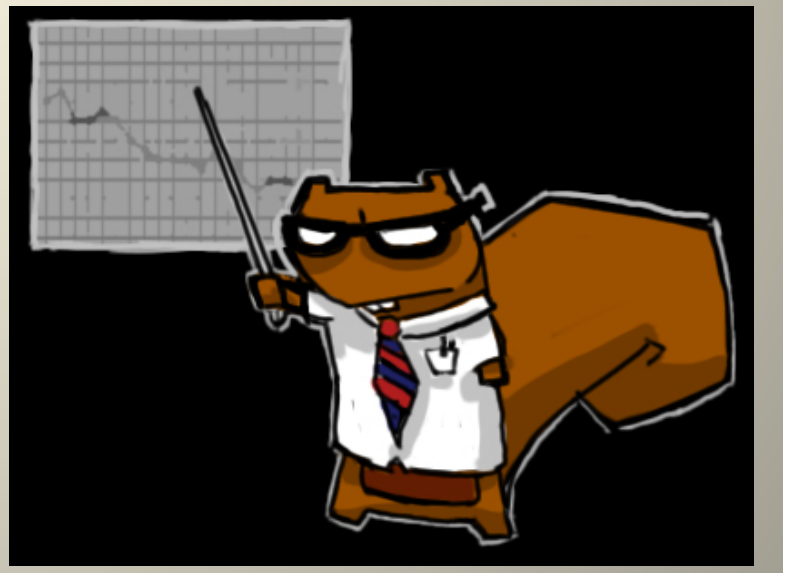
Selection Bias: Examples

- Researchers - Choose particular products or vulnerability types to research
- Vendors - Conduct internal research based on internal priorities; work with external researchers
- VDBs - Monitor limited set of disclosure sources

(Natural Selection Bias!)

Attrition Bias (tee hee)

- A type of selection bias
- During the study period, participants "drop out" and are not accounted for
  - E.g., in a diet study, people may drop out because they are not losing weight; participants at end of study show higher average weight loss
- Vuln stats often based on trending and vuln research dropout changes rapidly



Attrition Bias: Examples

- Researchers - Stops publishing new vulns or shifts to publishing a different vuln type. Stuff gets “too hard” for many researchers, never publish high-end vulns.
- Vendors - If a product reaches end-of-life
- VDBs – Stop monitoring idle source that resumes publishing. Stops monitoring new sources as carefully.

Sampling Bias

- “a non-random sample of a population, causing some members of the population to be less likely to be included than others, resulting in a biased sample “
- Which products are covered?
- Which vulnerabilities are covered?

Everyone excludes the poor lamprey.

Sampling Bias: Examples

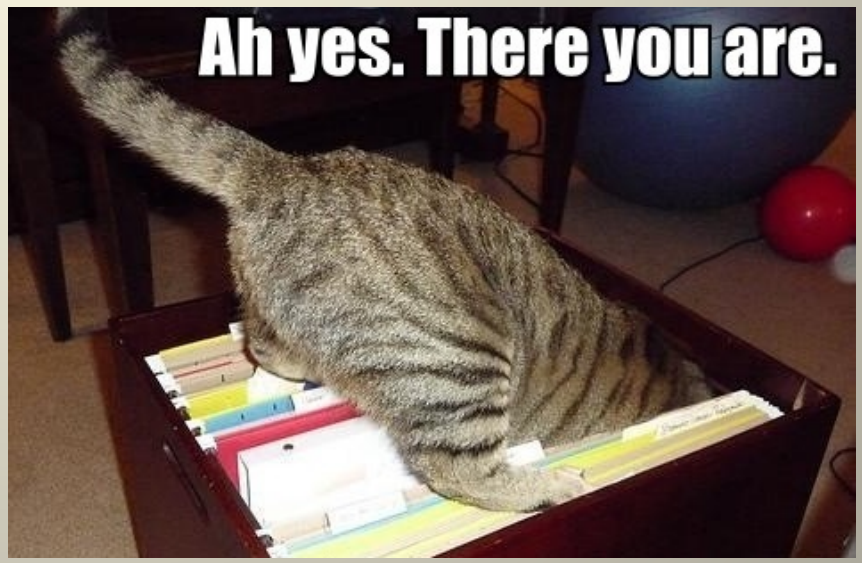
- Researchers – Because it's not a vulnerability we know how to find (e.g. skill)
- Vendors – Because it's a low-risk issue (e.g. path disc)
- VDBs – Because it's not a vulnerability at all (e.g. doesn't cross privilege boundary)
- The above a.k.a. "exclusion bias"

Because it's not an animal?

Publication Bias

Publication Bias Defined

- "The publication or nonpublication of research findings, depending on the nature and direction of the results." (Wikipedia)
  - Positive results bias: "authors are more likely to submit positive results"
  - File drawer effect: "many studies … may be conducted but never reported"



Publication Bias: Examples (Positive Results)

- Researchers
  - Only publish for high-profile products
- Vendors
  - Only publish patched, high-severity issues for supported products & versions
- VDBs
  - Only publish “verified” issues of a certain severity for “supported” products

Publication Bias: Examples (File Drawer Effect)

- Researchers
  - Don't publish low-risk or "lame" vuln types
  - Some won't publish at all (e.g. legal threats)
- Vendors
  - Don't publish low-risk or internally-discovered issues
- VDBs
  - Don't publish site-specific issues

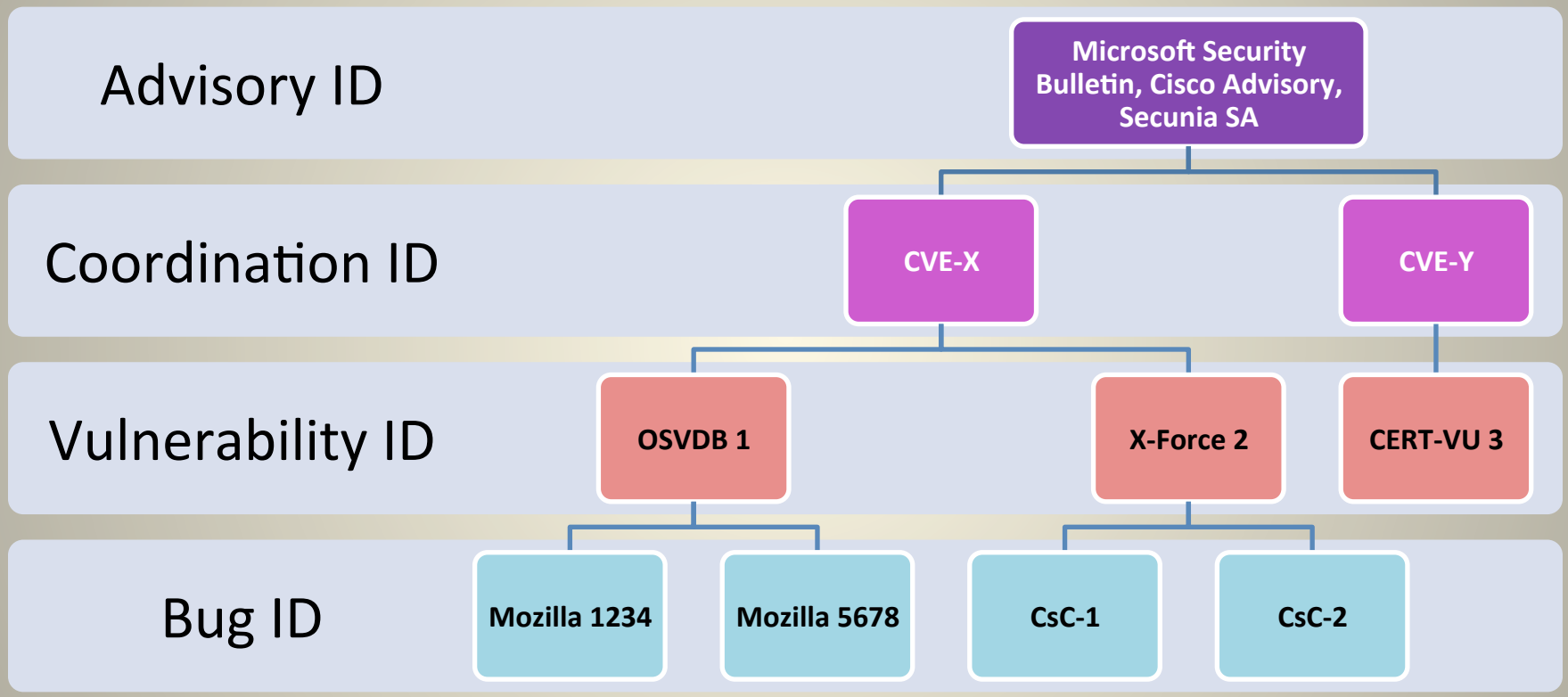
No one reports on me =(

Abstraction Bias

Abstraction: Units of Measurement (Vuln Stats' Achilles Heel)

- Advisories
- Patches
- Admin actions
- Vulnerabilities
- Coordination IDs
- Bug IDs

Different Audience → Different Abstraction



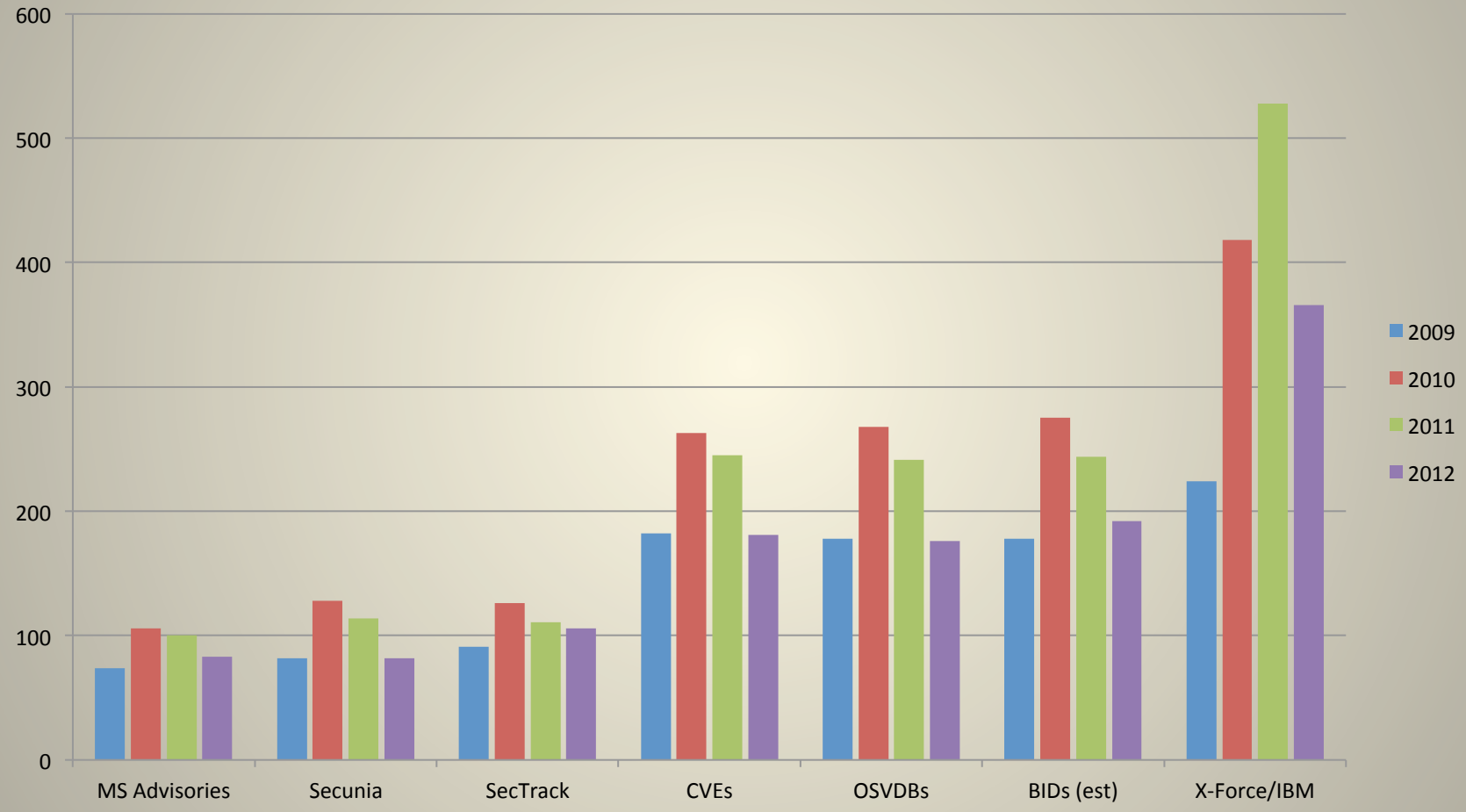
- *CVE was always intended as a coordination ID*
- *We originally thought that coordination could operate at the vulnerability level*
- *But, there's too much fluctuation and variation in vulnerability information in the early stages, when coordination ID is most needed*

Abstraction Bias: Examples

- Researchers - Release many advisories for one core issue, boosting counts.
- Vendors - Combine many vulns into the same advisory for sysadmin convenience. Bundle silent security fixes into non-security updates.
- VDBs - Uses the level that is best for the intended audience and balances analyst workload.

Kiwi abstraction is much cuter.

Counting Differences from the same set of Microsoft Bulletins



"Based on my count, there were 83 vulnerabilities announced by Microsoft [in 2012]" – Alex Horan, CORE Security

Measurement Bias

Reliability of the Measuring Device

- Reliability refers to how consistently a measuring device is applied.
  - The “measured” value might not be the “true” value
- If the discloser = the measuring device… (Hahaha)
- In the vuln world, there is no such thing as a “thermometer” that always yields the same result when applied to the same software
  - Different researchers on the same product yield wildly different answers
  - Automatic code analysis is… well… not perfect

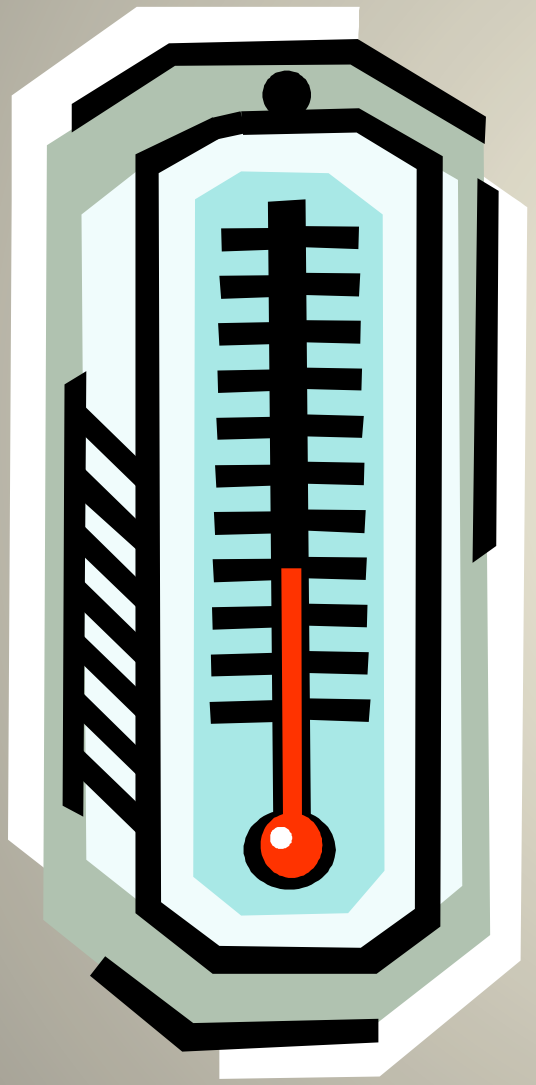
Measurement Bias: Examples

- Researchers: Over-estimate severity, or do not validate findings
- Vendors: Under-estimate severity, obfuscate vulnerability types
- VDBs: Misinterpret or completely miss external disclosures

More than 90 percent of the vulnerabilities disclosed are moderately or highly critical – and therefore relevant. (NSS Labs)

"There is one animal!" *BZZZT* →

CVSS*: **Everyone's** Favorite Thermometer



10.0 – ** CYBER POMPEII

(or completely unspecified)

8.0 – 8.9 – rarely seen in the wild

7.0 – max possible application score (Oracle, Apache, etc.); max local compromise

6.4 – Oops, I broke the Internet (Kaminsky)

5.x –remote full path disclosure, local read ALL non-root files, memory address leak

4.9 – local kernel crash 'n' burn

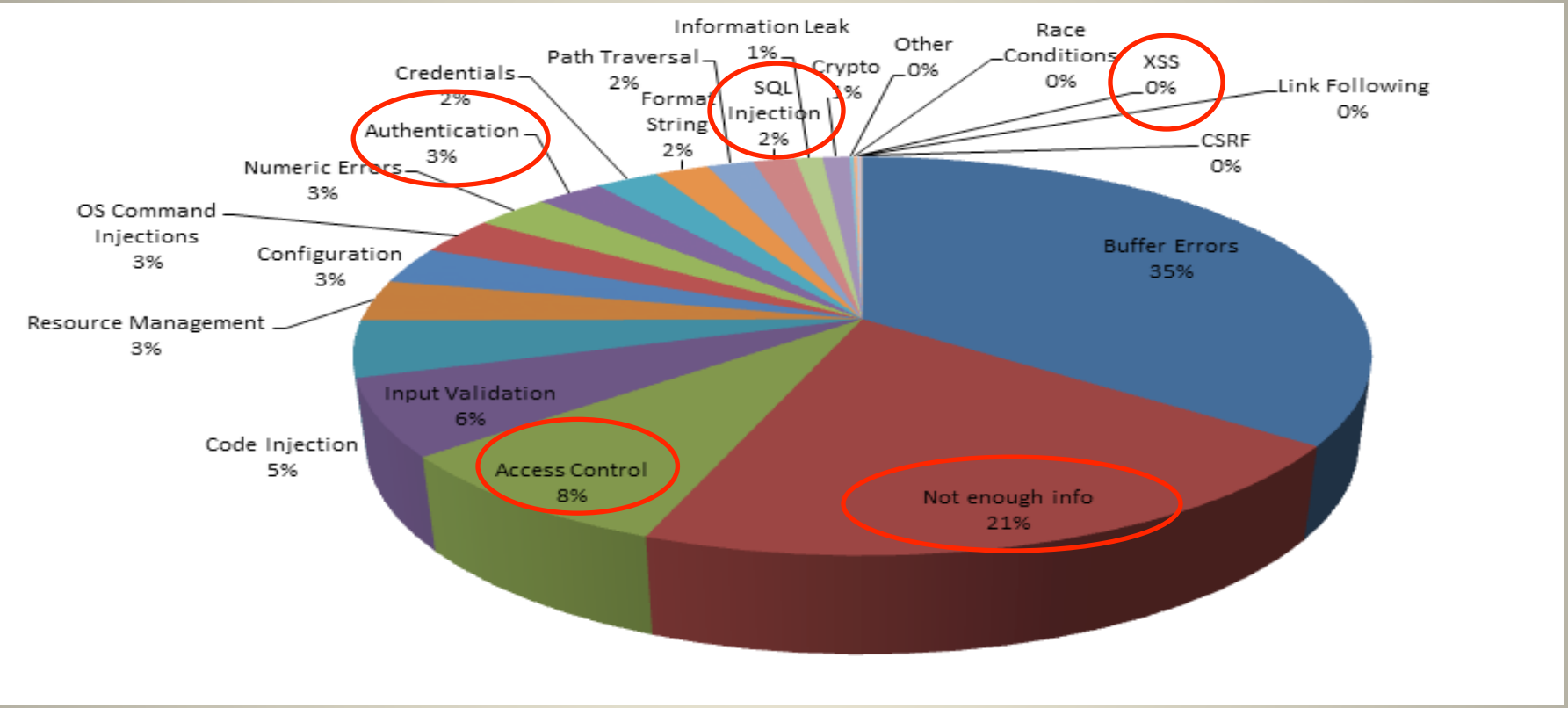
4.3 – typical XSS maximum

2.x – would YOU publish an advisory for this one? No. No, you wouldn't.

** creepy @alexhutton might say: "not endorsed by 9 out of 10 Bayesians"*

- Scoring is not done consistently
- Only scores impact to "IT asset" (in v2, intended as the host OS)
- Formalizes selection bias ("CVSS >= 7.0")
- CVSSv3 to make improvements

CVSS Measurement Bias "In the Large"



"Selecting only critical vulnerabilities (CVSS score of 10) yields additional significant information."
– Yves Younan, Sourcefire

- "Not enough info" often → CVSS 10.0
- XSS and SQL injection usually score less than 7 in NVD
- Varying levels of abstraction in vuln types

Chaining Bias

- When multiple forms of bias influence each other, cascades down to statistics:
  - Researcher chooses product, publishes
  - Vendor ignores low-risk, only confirms high-risk
  - VDB may misinterpret disclosure, ignore low-risk, abstract differently (abstract based on researcher, vendor, or both)
- Worse, all of the above repeats in very different ways and combinations.

Disclaimer: Bias is Not Always Bad

- Different organizations = different focus
- Bias is OK, just qualify and disclaim it!

Sources of Bias

Researcher

- Skills
- Focus
- Disclosure

Vendor

- Prioritization
- Details

Vuln DB

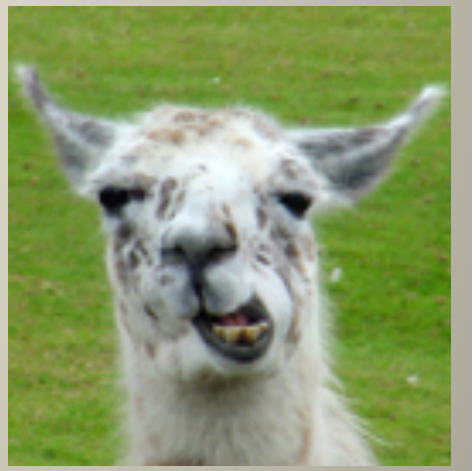
- Effort levels
- Monitoring
- Abstraction
- Selection processes *

Researcher Bias







"Disclose All the Vulns!"

Researcher Bias – Skills, Focus, Disclosure

- Skills and focus affect Selection Bias
  - Vulnerability types (skill to find)
  - Products (easy/cheap, or most-popular, or new class)
  - Fads (e.g. XSS wave, SQLi flood)
  - Productivity of solo vs. group
- Disclosure choices affect Publication Bias
  - Severity (code execution is sexy)
  - Disclosure channel (blog, mail list, direct to VDB?)
  - Vulnerability markets (e.g. ZDI, Bug Bounties)
- Disclosure choices affect Abstraction Bias
- Examples…

Notable Examples of Researcher Selection/Publication Bias

- Selection
  - Litchfield, Kornbrust, Cerrudo vs. “Unbreakable” Oracle
  - Month of (Browser|PHP|ActiveX|etc.) Bugs
- Publication
  - The Dino Dilemma: memory corruption experts wouldn’t dare publish an XSS
  - The Oberheide Oversight: not publishing can cost you $10K

The Four I's of Measurement Bias

- Incomplete
  - Missing versions, product names
  - Missing patch information
- Inaccurate
  - Incorrect diagnosis
  - Blatantly wrong
- Inconsistent
  - Acknowledgement discrepancies
  - Bug type discrepancies
  - Varying severities
- Incomprehensible
  - Poor writing
  - Lack of clear formatting

Coordinated disclosure between researcher and vendor frequently wipes these out.

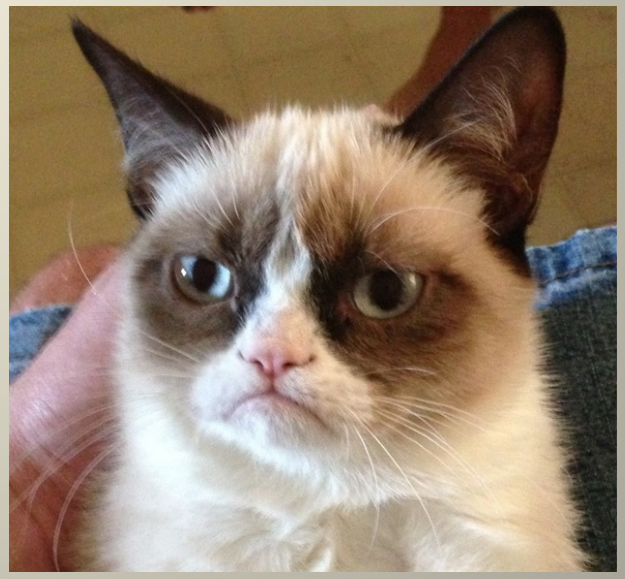
The r0t Method of Vuln Analysis

- Be a teenager, with plenty of spare time
- Go to a software repository web site
- Download a package or try its demo site
- Do blatantly simple SQL injection and XSS:

```
'
```

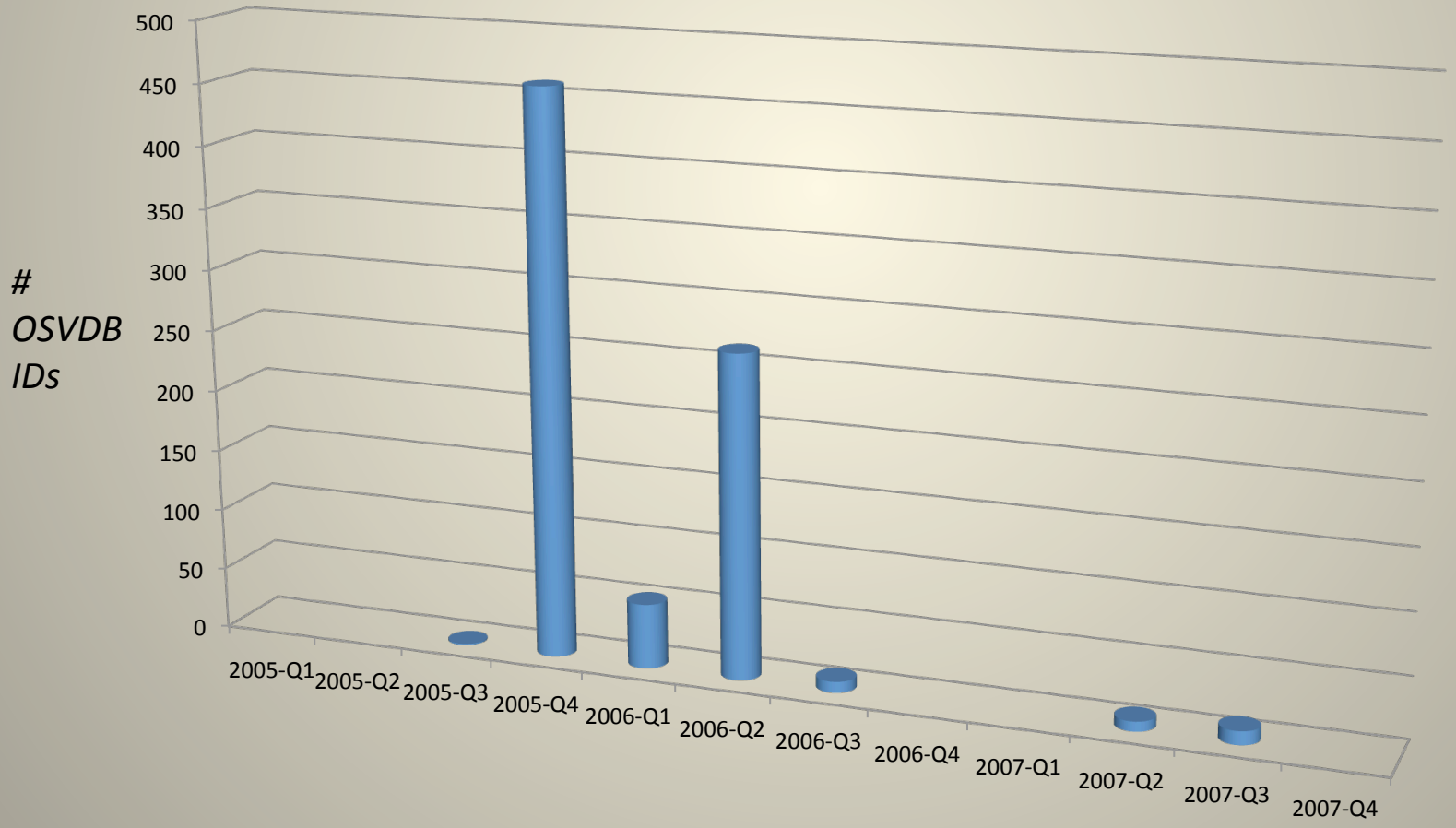
```
<script>alert('XSS')</script>
```

- Move on after 10 minutes
- Disclose the issue on your blog
- Mail all the VDBs

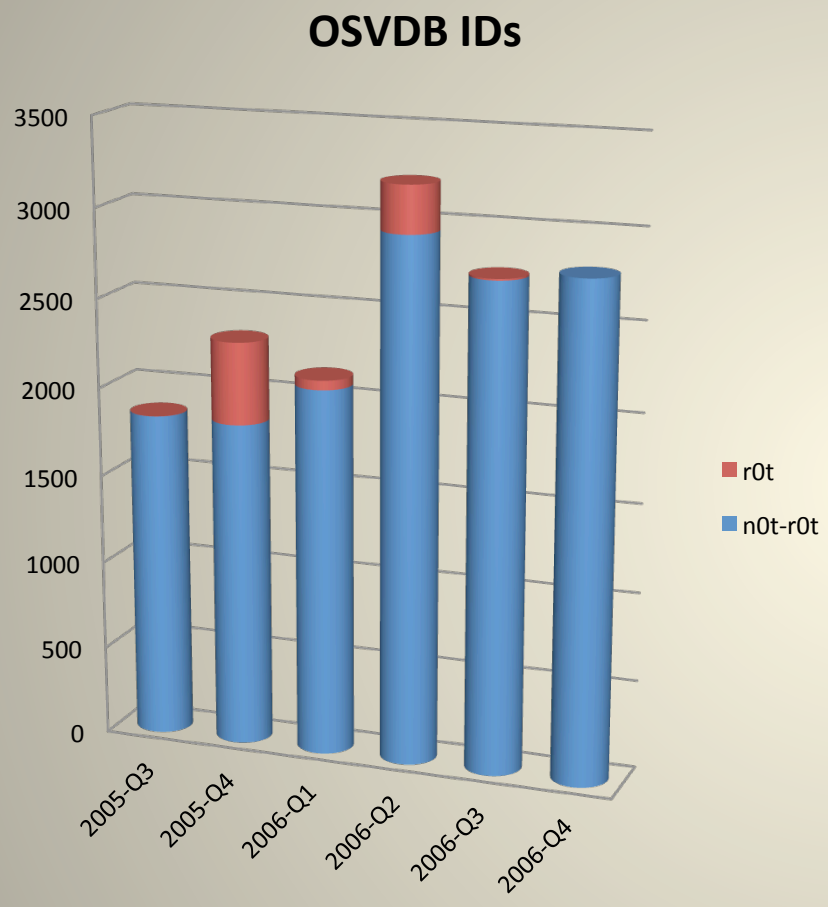


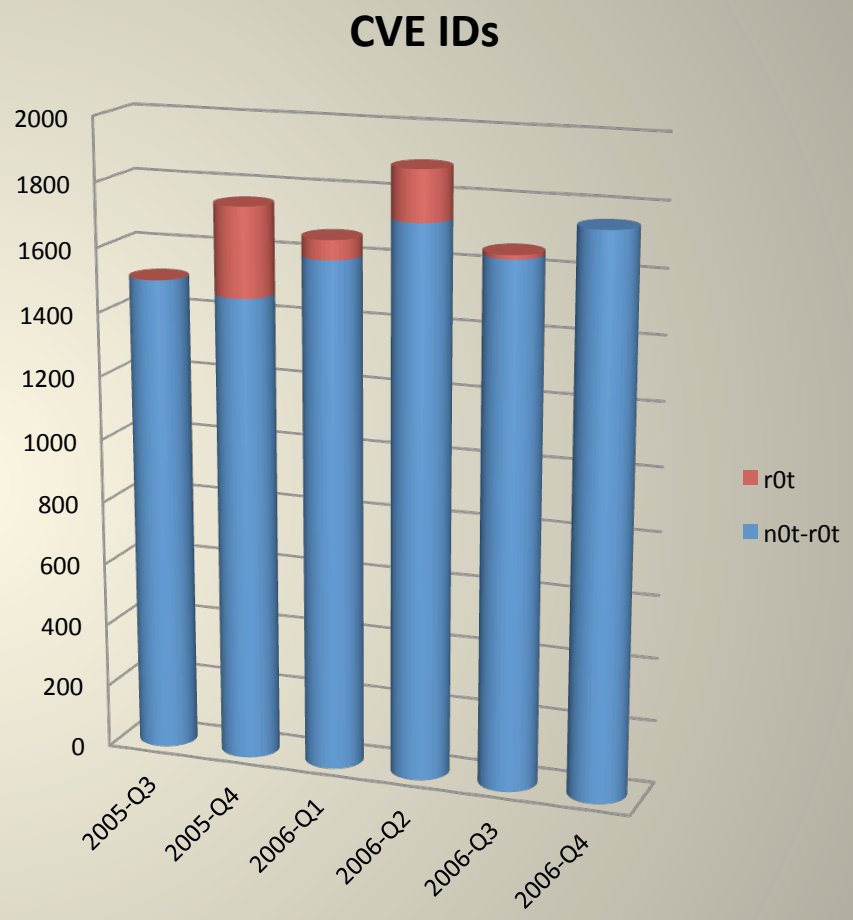
The r0t Method of Vuln Analysis

- Is it successful? YES
- 810 vulnerabilities disclosed
- Between 2005-08-09 and 2010-09-16



The r0t Speed Bump: Quarterly IDs



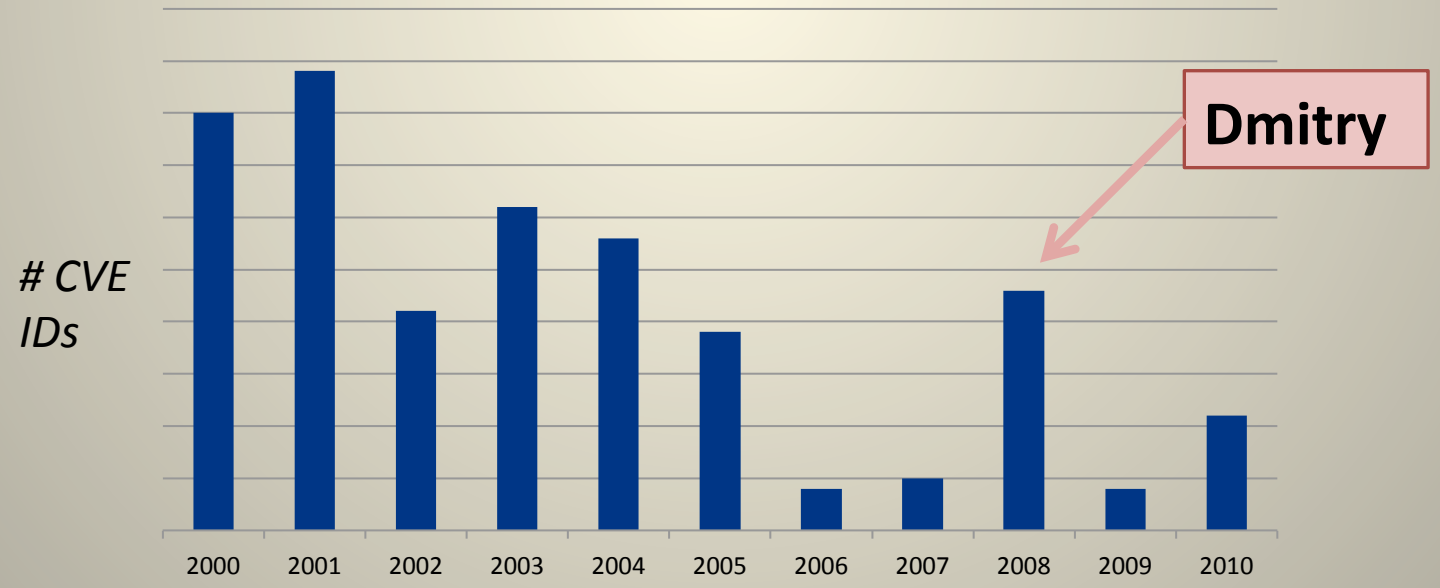


"Private hackers are more likely to use techniques that have been circulating throughout the hacker community. While it is not impossible that they have managed to generate a novel exploit to take advantage of a hitherto unknown vulnerability, **they are unlikely to have more than one**." -- Martin C. Libicki (RAND) 2009

Grep-and-Gripe: Revenge of the Symlinks

```
grep -A5 -B5 /tmp/ $PROGRAM
```

- Dmitry E. Oboukhov, August 2008
- Run against Debian packages
- This kind of thing really hurts pie charts of different vulnerability types

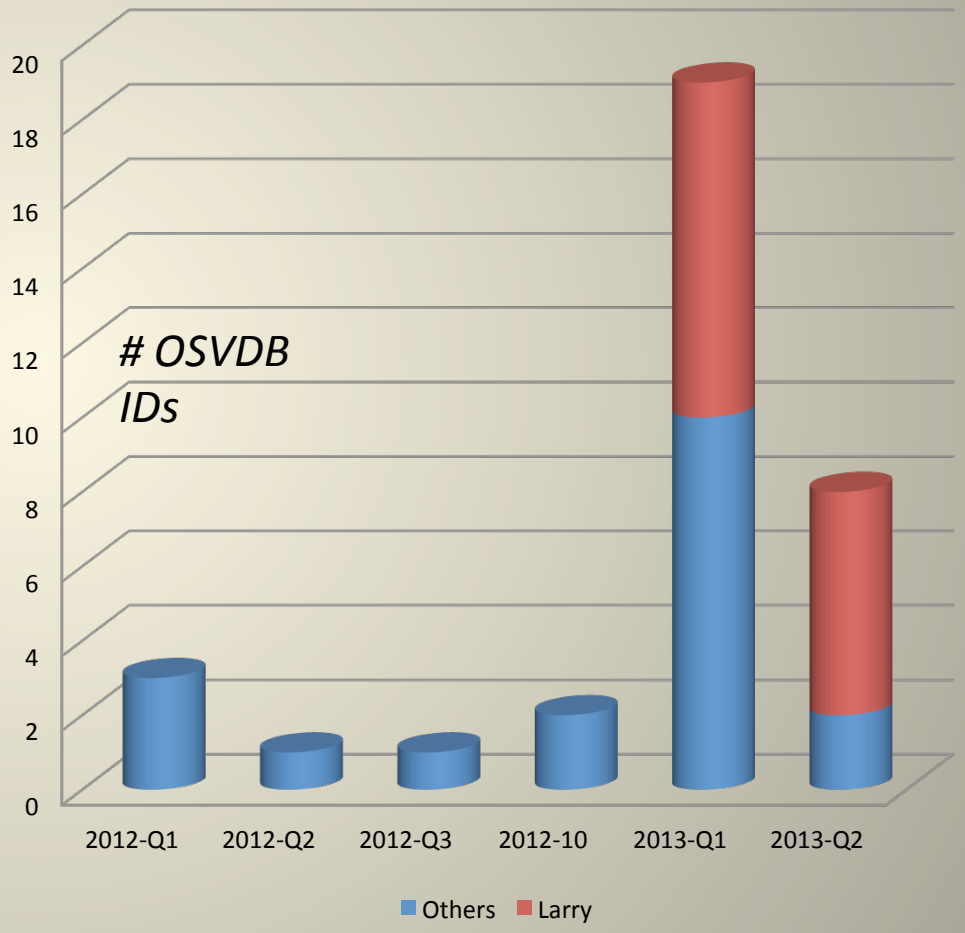


Raw number of symlinks reported over time (CVE)

Grep-and-Gripe 2: Larry Cashdollar*

* That's his real last name. He swears it!

- Grep-and-gripe
- Old-school symbolic links and context-dependent OS command injection
- Those are dead, right?
- Enter Ruby Gems



Grep-and-Gripe 3: Attack of the Clones

(aka, “Why False Positives Suck” or “Measurement Bias”)

```
# grep "include.*\$"
```

abc.php

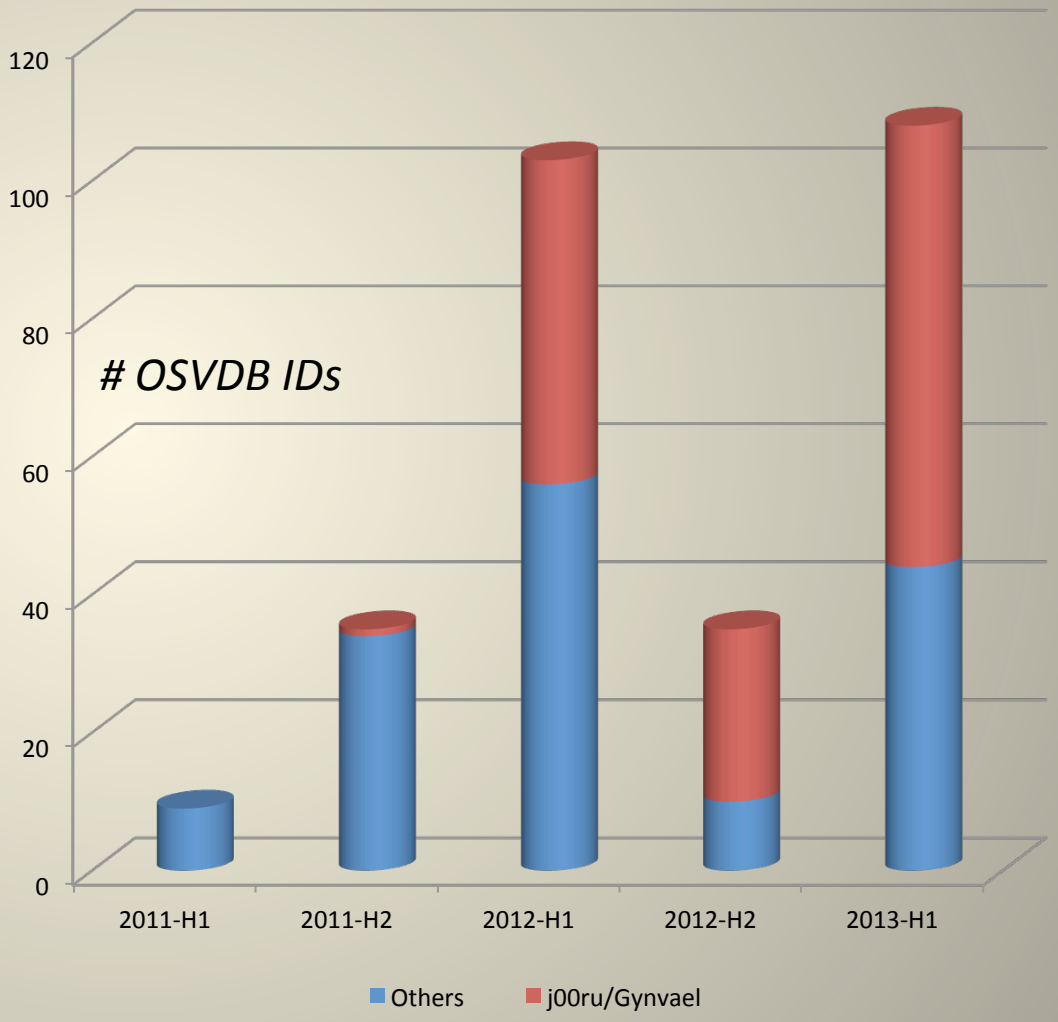
```
$language = “english”;
…
include(“$language.php”);
```

http://example.com/abc.php?language=[RFI]

VDBs

FFmpeg

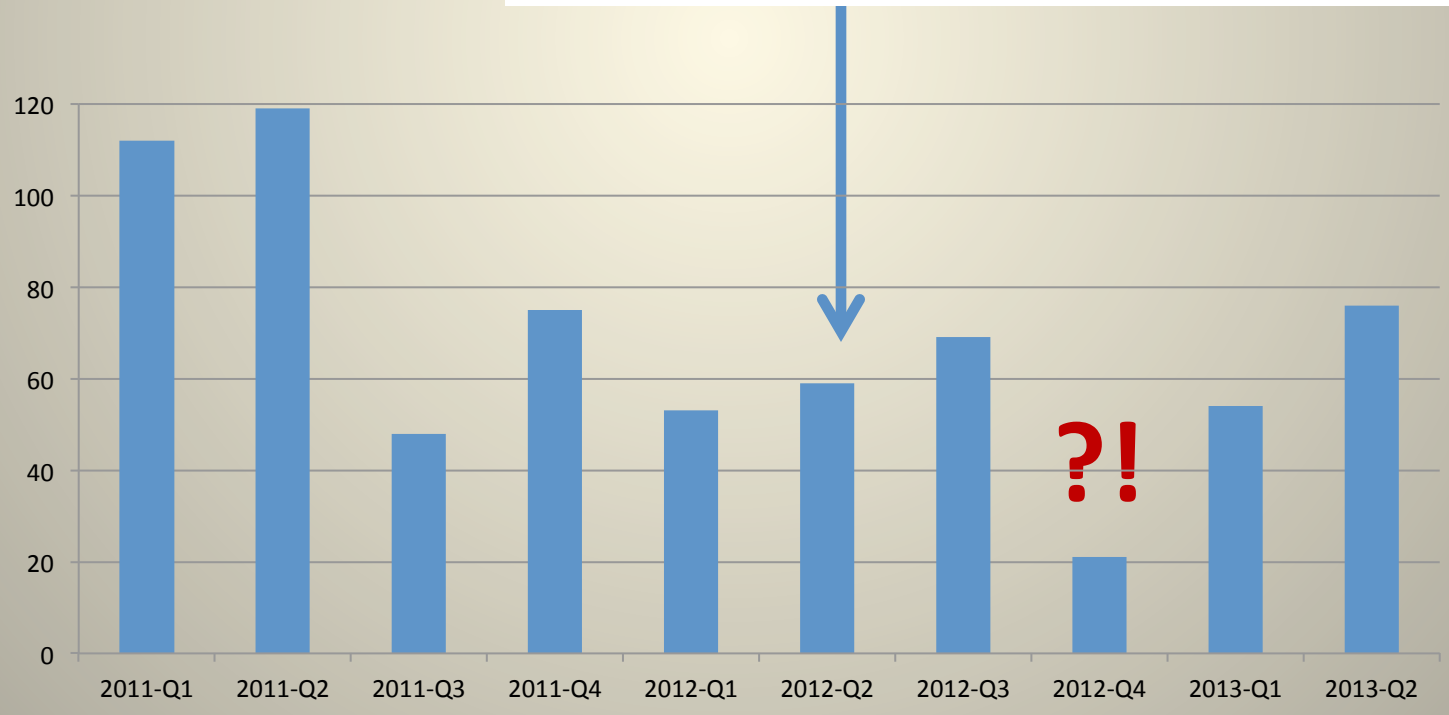
- Number of vulns skyrocketed recently
- Maybe because of who was looking at it?



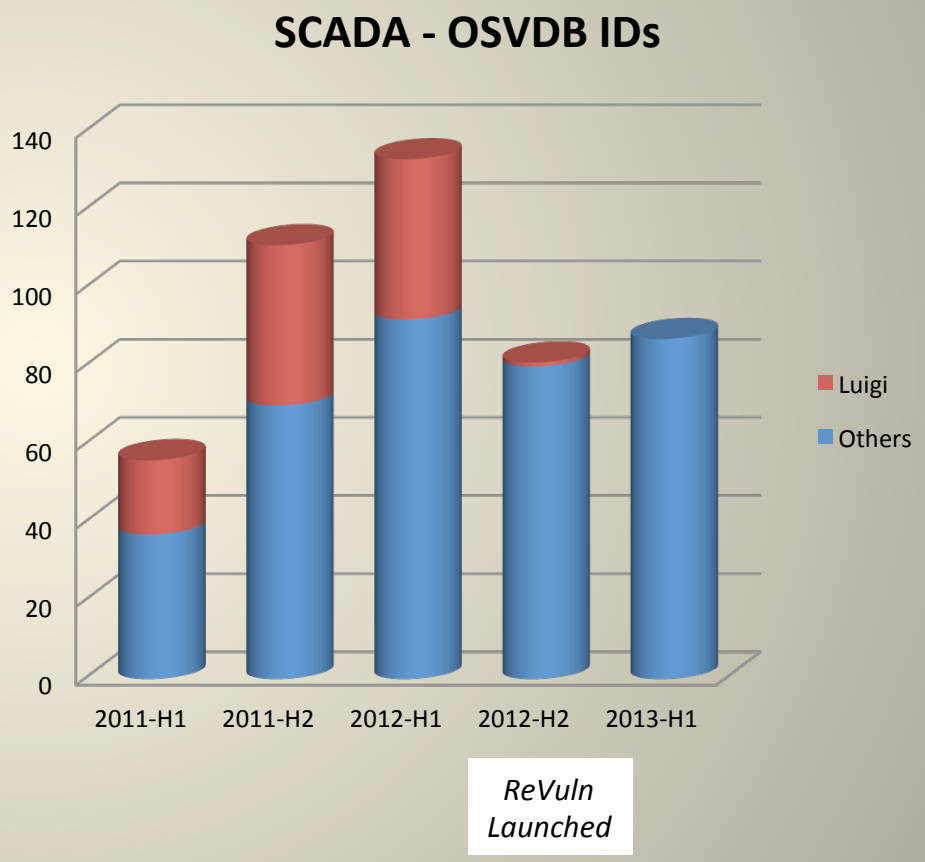
In 2012, FFmpeg is listed as the #5 vendor with 6% of "*highly critical, easy to exploit vulnerabilities*" (NSS Labs)

Researcher Attrition Bias: ZDI

"VCP and ZDI reversed their five year long rise with a reduction of more than 50% of vulnerability disclosures in 2012… [this] correlates with reports of the vulnerability and exploit market rapidly expanding" – NSS Labs



The Luigi Lossage: Selection & Publication Bias



* 2011 Luigi stats may be higher than shown.

Abstraction (a.k.a MustLive Mess)

- Finds one vulnerability, then over time does an advisory for each software he finds that bundles it.
- How do you count it? Some VDBs abstract per software! Some add based on root-cause.

Jon Rohan	ZeroClipboard	1.0.7
YAML	YAML	4.0.2
Multiproject	Multiproject extension for Trac	1.4.21
Object Computing, Inc.	TAO	2.3.1
InfoGlue Community	InfoGlue	2.1
SpryMedia	TableTools Plugin for DataTables	2.1.4
mtstic100	User Collections Plugin for Piwigo	1.1.0
em-shorty	em-shorty	0.5.0
Max Bond	Code Insert Manager Plugin for WordPress	2.3.1
flashpixx	GeSHi Source Colorer Plugin for WordPress	0.13
Jaspreet Chahal	JC Coupon Lite Plugin for WordPress	2.1
Cleeng	Cleeng Plug and Go Plugin for WordPress	2.3.2
	PayPal Digital Goods Plugin for WordPress	2.2.13
Zoptm	Zoptm Live Chat Plugin for WordPress	1.2.5
Matthew Restorff	Buckets Plugin for WordPress	0.1.9.3
Prasanna SP	Tiny URL Plugin for WordPress	1.3.2
WP Academy	WP Clone Plugin for WordPress	2.1.1
PT Lemuel ColorLabs	MobileView Plugin for WordPress	1.0.7
Digital Telepathy	SlideDeck 2 Lite Responsive Content Slider Plugin for WordPress	2.1.20130306
blogjunkie	Click to Copy Grab Box Plugin for WordPress	0.1.1
tmatht.eu	BP Code Snippets Plugin for WordPress	2.0
steamdev.com	zClip	1.1.1
bytesforall	Montezuma Theme for WordPress	1.1.7
PremiumPress Limited	CouponPress Theme for WordPress	7.1.3
kaptinlin	Striking Theme for WordPress	5.1.9.5
whoathemes	Black and White Theme for WordPress	1.5
CloudBees, Inc.	Jenkins	1.502
	Jenkins Enterprise	1.466.13.1
		1.480.3.1
	Jenkins LTS	1.480.3
Search and Share Plugin	Search and Share Plugin for WordPress	0.9.3

- ZeroClipboard zeroclipboard.swf id Parameter XSS

Fuzzmarking – Generating Good Data

- Kaminsky, Cecchetti, Eddington (2011)
- Used the same fuzzer against Windows Office / OpenOffice, and PDF viewers for software from 2003, 2007, 2010
- Minimized bias:
  - Selection bias: used same environments and time frames
  - Measurement bias: use same tools, normalize counting unique crashes using !exploitable
  - Abstraction bias: use results from same tools
  - Publication bias: raw data provided?
- Methodology shared: Yes!

(Honey)Vendor Bias

Vendor Bias

- Vendor advisories vary greatly:
  - Customers-only or public?
  - Vulnerability vs Stability vs Hardening?
  - Vulnerability details or vagueness?
    - Oracle/HP - CVSS only essentially!
    - Microsoft - in the middle with "memory corruption“
    - Red Hat – Vuln types, access to most bug reports
    - Linux Kernel - diffs (undiagnosed vuln types/vectors)

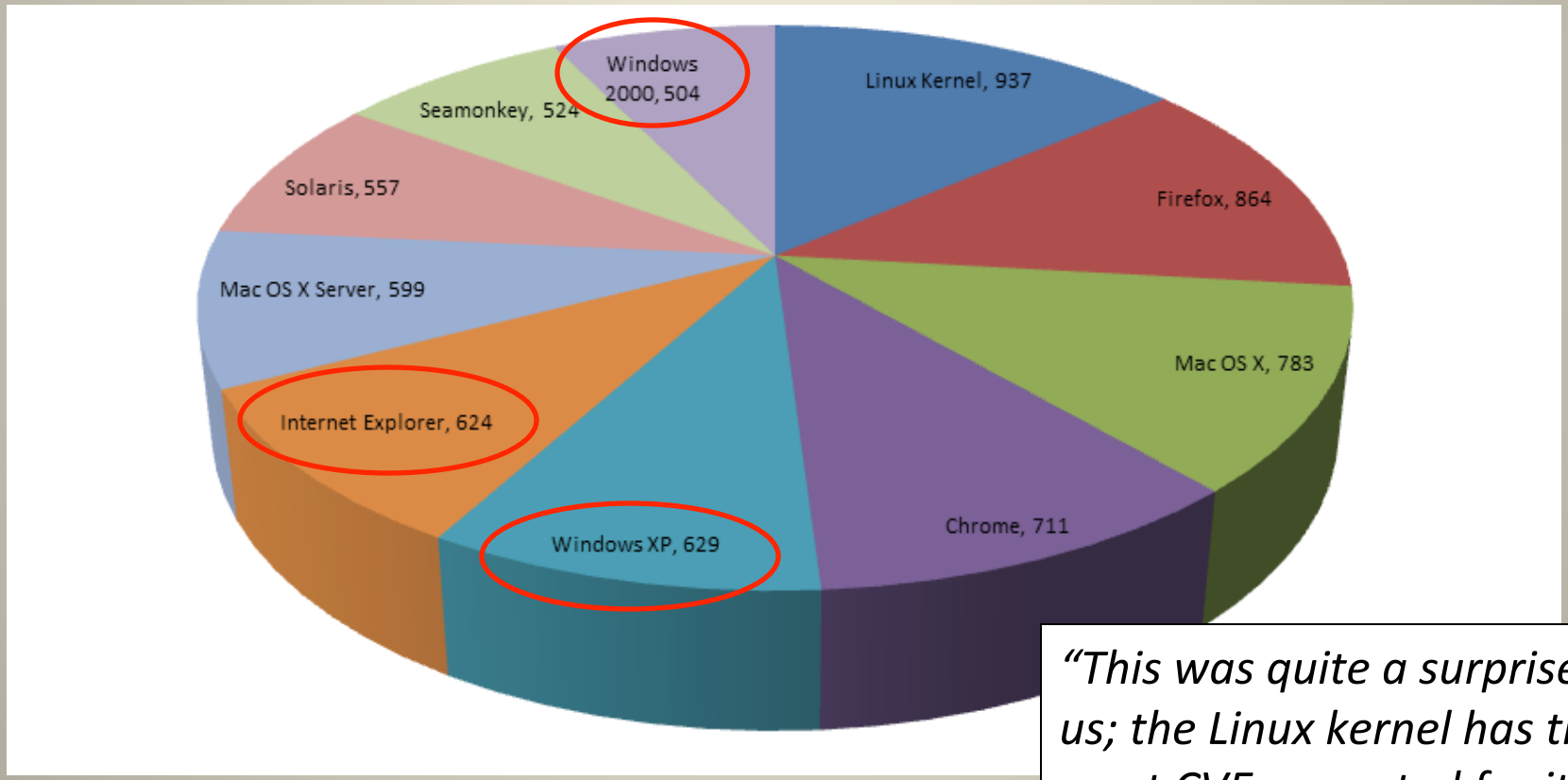
Vendor Publication Bias

- No public access to vulnerability advisories
  - E.g., SAP, Juniper (until mid-2013), Google (no real "advisories" for most products)
- Not reporting self-discovered vulnerabilities
  - E.g., Microsoft, Linux kernel team
- Not reporting low-severity issues
  - Unless rolled up with higher-severity

Smug ~~dog~~ vendor is **still** smug.

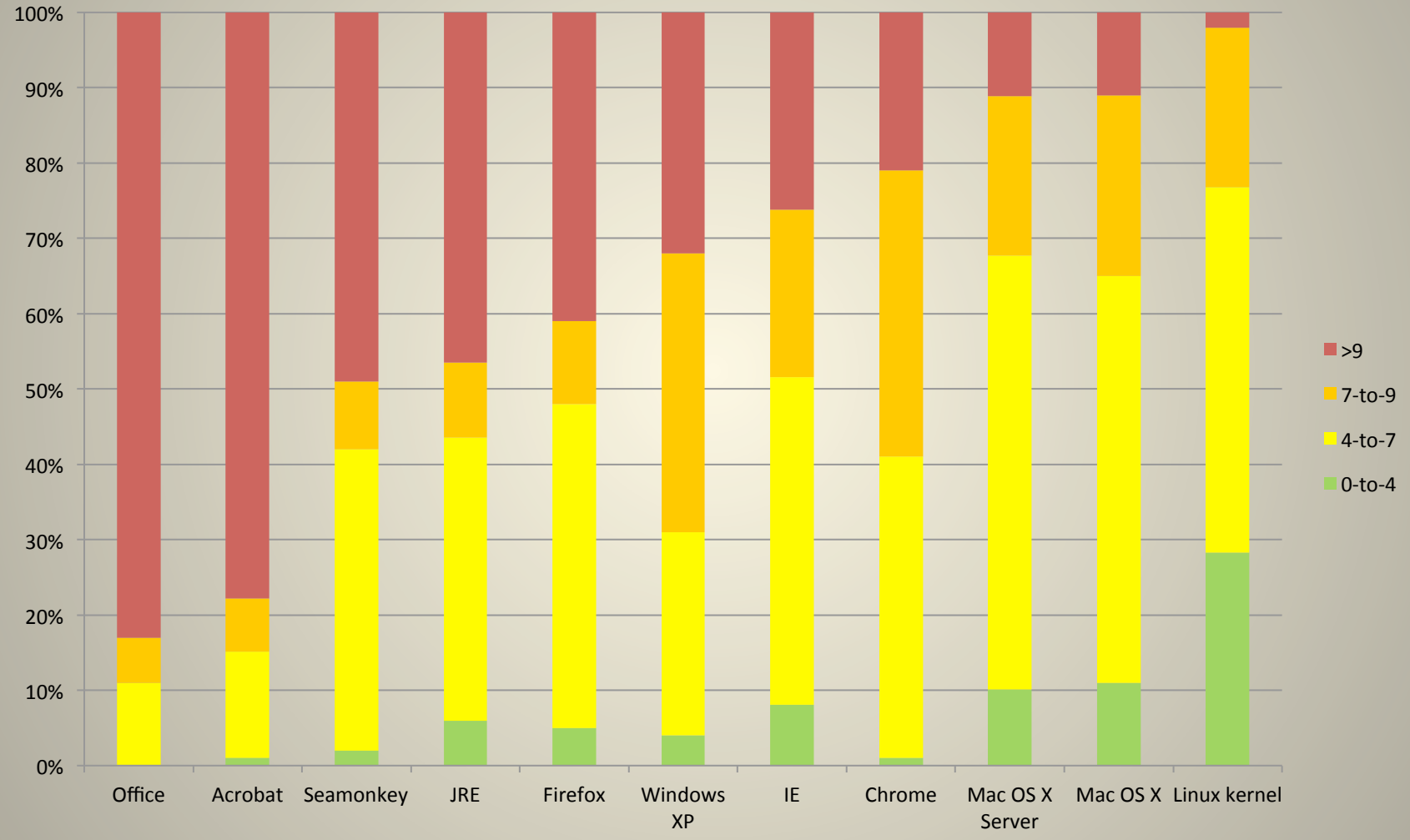
Publication Bias: Comparing Self-Disclosing and "Silent Patching" Vendors

"1. Google Chrome (76 reported vulnerabilities); 2. Apple Safari (60); 3. Microsoft Office (57)" – Bit9, 2010



"This was quite a surprise to us; the Linux kernel has the most CVEs reported for it"
- Sourcefire, 2013

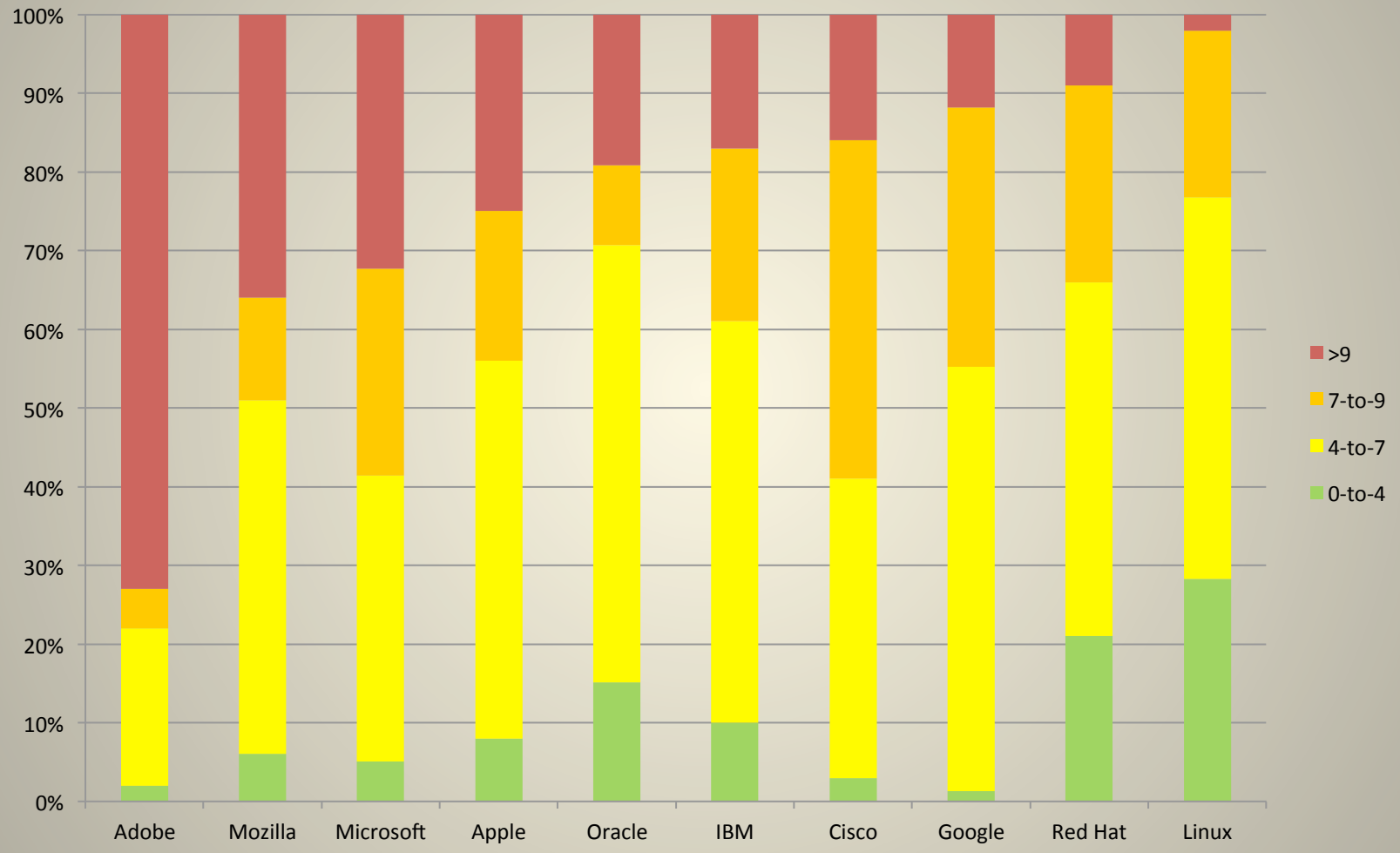
CVSS Score Distribution - Some Popular Products (NVD-based)



- Desktop/browser apps with high-severity issues (CVSS bias)
- Linux kernel with 28% low-severity issues, only 2% 9+

** Numbers from www.cvedetails.com*

CVSS Score Distribution – Some Popular Vendors (NVD-based)

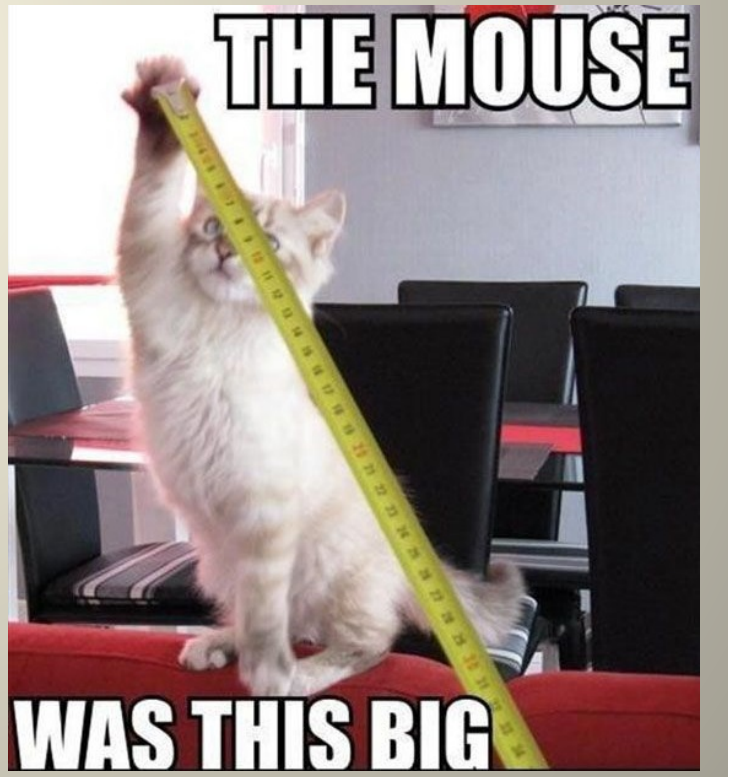


- Browser-heavy vendors with high-severity issues (CVSS bias)
- Linux kernel with 28% low-severity issues, only 2% 9+

** Numbers from www.cvedetails.com*

Measurement Problems by Vendors

- [Under|over]-estimating severity of issues
  - "That can't be exploited"
  - "That's just the way the Internet works!"
  - Memory corruption "catch-all"
- Insufficient technical details
  - Frustrates version & vuln type
- Lack of cross-references to other disclosures
  - Increases risk of duplicates
- Lack of commentary on 0-days



Vendor Abstraction Bias

- First abstraction by customer action (patch)
  - Minimize frequency of “alerts”
- Second abstraction by “vulnerability” (CVE)
- Some may abstract by code fix:
  - SQL injection and XSS might be fixed with a single “input validation” patch that converts an input to an integer
  - If vendor doesn’t publish how it was fixed, VDBs don’t know “root cause” and may abstract incorrectly

Don’t poke the hog. Or else.

Vendor Measurement Bias and CVSS

"What these vendors [Oracle and HP] lack in quantity of vulnerabilities, they make up for in severity." - Yves Younan , Sourcefire #derp

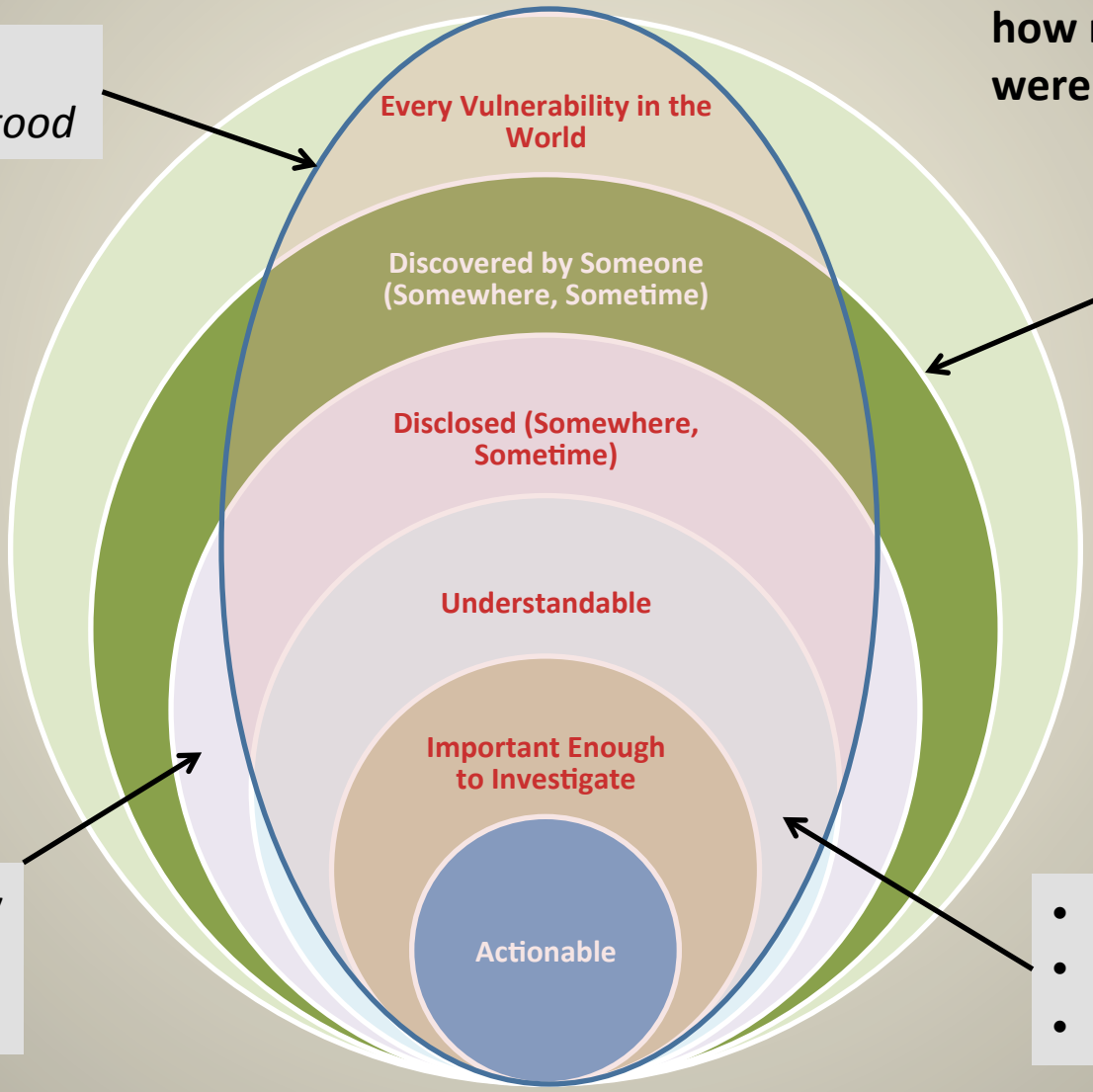
- Coincidence that for years, these two companies didn't publish any details?
  - CVSS 10.0
- Oracle also (now) owns Java, which can get many 10.0's because of CVSS "Windows user running as admin" scoring
  - Same as Adobe Flash

Adobe is listed as the #1 vendor in 2012 with 23% of "highly critical, easy to exploit" vulnerabilities" – NSS Labs

VDB BIAS

The World of Vulnerability Disclosure (Not to Scale)

FACT: No VDB knows how many vulnerabilities were disclosed.



Actually real and well-understood

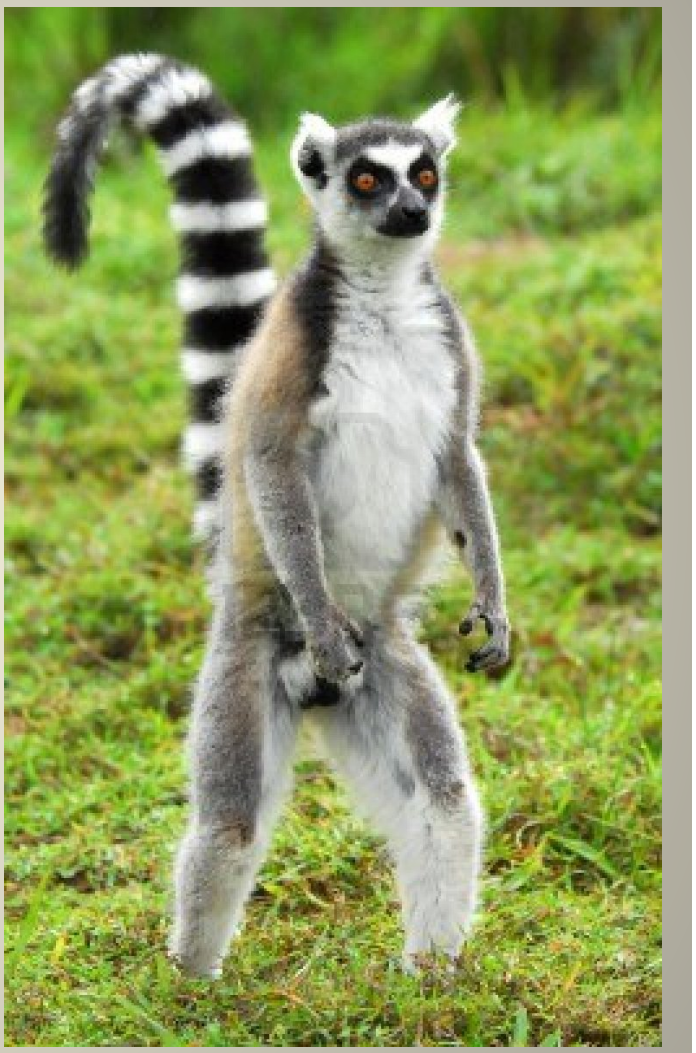
0-days

- *Customer-only*
- *Obscure blogs*
- *Bug reports*

- *Language barriers*
- *Writing quality*
- *Relevant details*

Intentional Selection Bias by VDBs

- Products covered
- List of sources monitored
  - Catering to customers
- Severity / importance of vulns
- Disclosure confidence
  - Bad researchers can be blacklisted or de-prioritized
- Patch or workaround availability
- Site-specific [out of our scope]
- Time of Disclosure
  - All or some? Going how far back?
- Changelog hunting in OSVDB
  - “hardening” vs. “vulnerability”



COME AT ME BRO!

Unintentional Selection Bias in VDBs

- Raw volume of disclosures
- List of monitored sources
  - Resource limitations
  - Staffing levels (up and down)
- Staffing expertise
  - Less experience implies more editing/review
- False positives (i.e., erroneous reports)
  - Related to analytical effort
- Overhead beyond content generation

:(

VDB Publication Bias: Criteria for Inclusion (or Exclusion)

- Undisputed
- Vendor Disputed
- Third-party Disputed
- Third-party Verified
- Self-verified
- Fix/workaround available
- Not a Vuln
- Myth / Fake

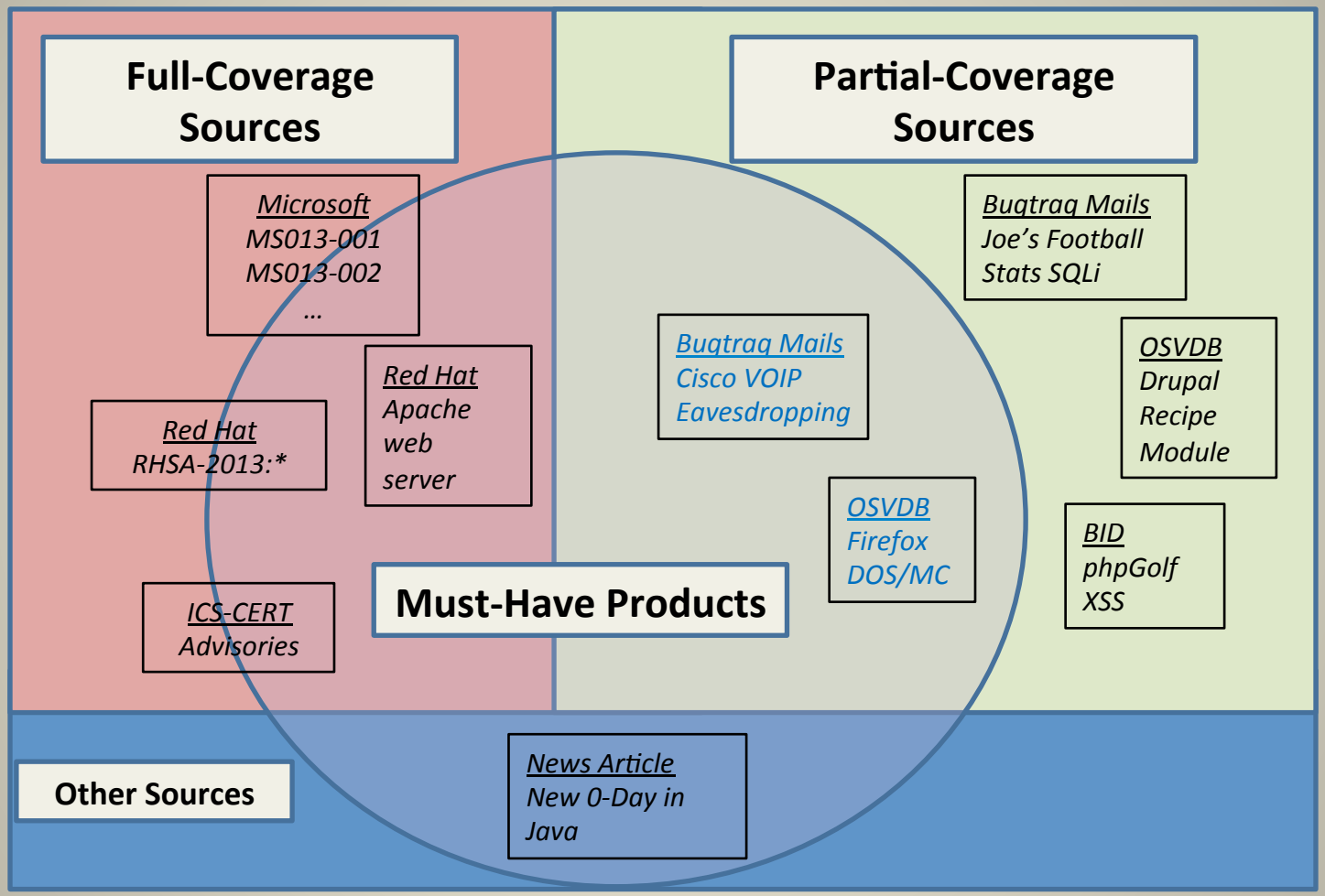
What do I have to do with VDB publication bias? NOTHING!

Types of VDB Coverage

- Targeted or Specialty
  - Concentrates on a subset of vulnerabilities, typically the highest priority to the VDB's intended audience (aka "customers")
  - E.g. CVE Sources/Products list

- Comprehensive
  - Tries to capture all publicly-disclosed vulnerabilities
  - Still variation due to different sources monitored

Selection Bias in Action: CVE's Sources & Products List



- Full-Coverage Source: Every advisory published by this source must receive a CVE.
- Must-Have Product: Every advisory published about this product must receive a CVE.
- Partial-Coverage Source: some advisories should receive CVEs (especially for Must-Have products).

CVE Sources / Products

As of September 2012

Full-Coverage Sources

- Adobe
- Apache Software Foundation: Apache HTTP Server
- Apple
- Attachmate: Novell
- Attachmate: SUSE
- Blue Coat - kb.bluecoat.com
- CA - support.ca.com
- Check Point: Security Gateways product line (supportcenter.checkpoint.com)
- Cisco: Security Advisories/Responses
- Citrix - support.citrix.com
- Debian
- Dell Desktop/Notebook product lines
- Dell SonicWALL Network Security product line - Service Bulletins
- EMC, as published through Bugtraq
- F5 - support.f5.com
- Fortinet FortiGate product line (kb.fortinet.com)
- Fujitsu Desktop/Notebook product lines
- Google: Google Chrome (includes WebKit)
- HP: Security Bulletins
- IBM: issues in IBM ISS X-Force Database
- Internet Systems Consortium (ISC)
- Juniper: juniper.net/customers/support (JunOS?)
- Lenovo Desktop/Notebook product lines
- McAfee - kc.mcafee.com
- Microsoft: Security Bulletins/Advisories
- MIT Kerberos
- Mozilla
- OpenSSH
- OpenSSL
- Oracle: Critical Patch Updates
- RealNetworks (real.com)
- Red Hat
- RIM/BlackBerry- blackberry.com/btsc
- Samba Security Updates and Information
- SAP - scn.sap.com/docs/DOC-8218
- Sendmail
- Sophos - sophos.com/support/knowledgebase
- Symantec: Security Advisories
- Ubuntu (Linux)
- VMware
- Websense - websense.com/content/support.aspx
- HP: TippingPoint DVLabs
- HP: TippingPoint Zero Day Initiative
- ICS-CERT: ADVISORY
- MITRE CNA open-source requests
- US-CERT: Technical Cyber Security Alerts
- VeriSign iDefense

Must-Have Products

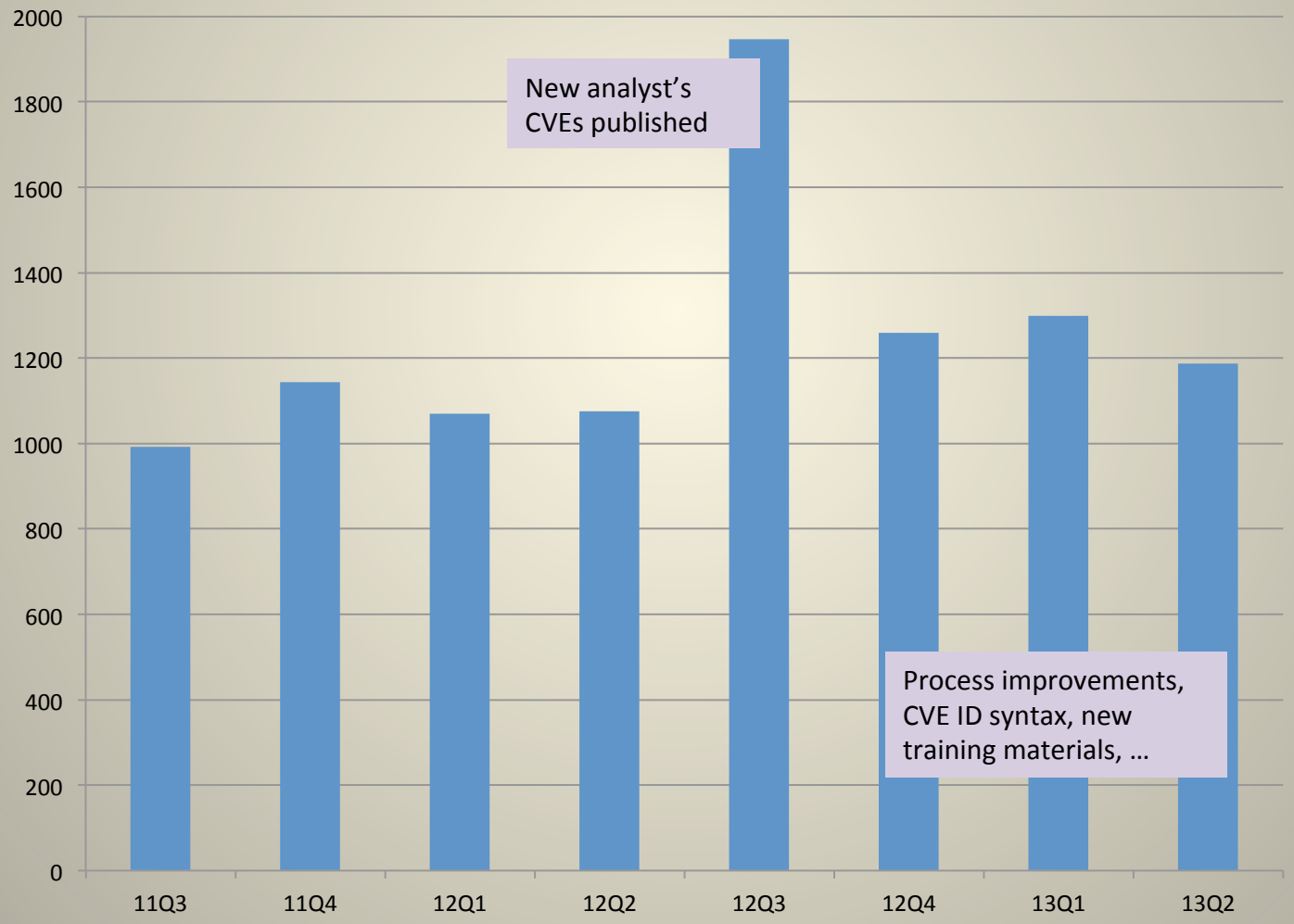
- Adobe: all
- Apache Software Foundation: All
- Apple: all
- Attachmate: Novell
- Attachmate: SUSE
- Blue Coat: all
- CA: all
- Check Point: Security Gateways product line
- Cisco: all
- Citrix - support.citrix.com
- Debian: all
- Dell: Desktop/Notebook product lines
- Dell: SonicWALL Network Security product line
- EMC: all
- F5: all
- Fortinet: FortiGate product line
- Fujitsu: Desktop/Notebook product lines
- Google: Google Chrome (includes WebKit)
- HP: all
- IBM: all
- Internet Systems Consortium (ISC): Bind
- Juniper: all
- kernel.org (Linux kernal)
- Lenovo: Desktop/Notebook product lines
- McAfee: all
- Microsoft: all
- MIT Kerberos: all
- Mozilla: all
- MySQL: all
- OpenLDAP: all
- OpenSSH: all
- OpenSSL: all
- Oracle:all
- PHP: core language interpreter
- RealNetworks:all
- Red Hat: all
- RIM/BlackBerry: all
- Samba: all
- SAP: all
- Sendmail: all
- Sophos: all
- Symantec: all
- Ubuntu: all
- VMware: all
- Websense: all

Partial-Coverage Sources

- Android (associated with Google or Open Handset Alliance)
- Apache Software Foundation: Apache Tomcat
- Apache Software Foundation: other
- CentOS
- Check Point: checkpoint.com/defense/advisories/public/summary.html
- Cisco: Release Note Enclosures (RNE)
- Drupal
- Fedora
- FoxIt Support Center - Security Advisories
- FreeBSD
- Gentoo (Linux)
- Google: other (not Chrome or Android)
- IBM ISS X-Force for non-IBM products
- IBM: issues not in IBM ISS X-Force Database
- Joomla!
- Juniper - JTAC Technical Bulletins
- kernel.org
- Mandriva
- NetBSD
- OpenBSD
- PHP core language interpreter
- SCO
- TYPO3
- WordPress
- attrition.org/pipermail/vim
- AusCERT
- Core Security CoreLabs
- DOE JC3 (formerly DOE CIRC and CIAC)
- Full Disclosure
- HP: TippingPoint Pwn2Own
- http://www.exploit-db.com/
- ICS-CERT: ALERT
- Juniper: J-Security Center - Threats and Vulnerabilities
- Microsoft: Vulnerability Research (MSVR)
- oss-security
- OSVDB
- Packet Storm
- Rapid7 Metasploit
- Secunia
- SecuriTeam
- SecurityTracker
- Symantec: SecurityFocus BugTraq (securityfocus.com/archive/1)
- Symantec: SecurityFocus Bugtraq ID (securityfocus.com/bid)
- United Kingdom CPNI (formerly NISCC)
- US-CERT: Vulnerability Notes

Selection Bias in Action: CVE Team Productivity

"The five year long trend in decreasing vulnerability disclosures ended abruptly in 2012 with a +12% increase" - NSS Labs



??

New Analysts Creating New CVEs

PSA

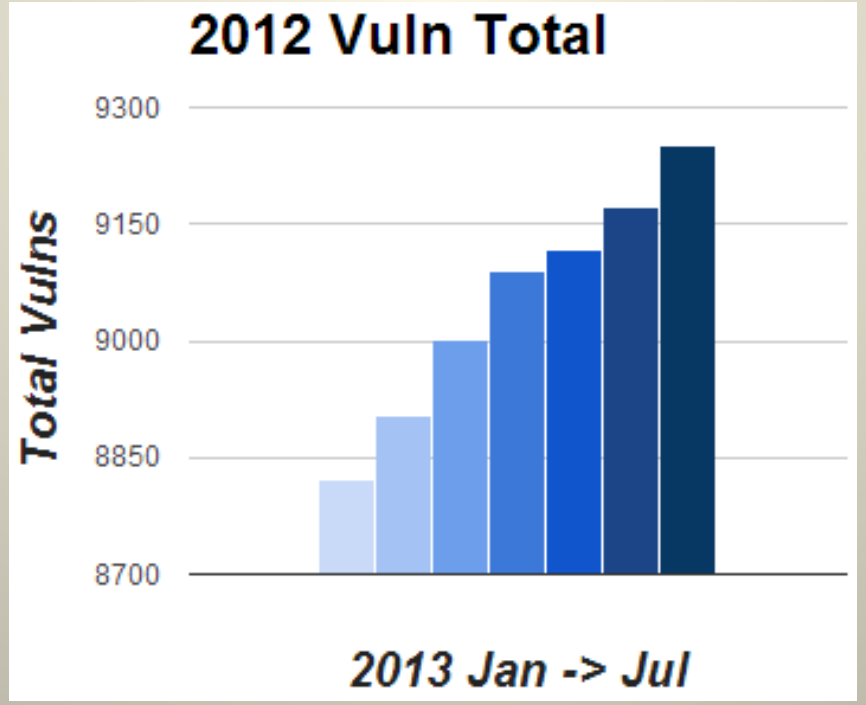
- Speaking of increased productivity…
- CVE ID syntax will change on January 1, 2014, to support more than 10,000 IDs in a year.

CVE-2014-1234
CVE-2014-12345 [5 digits if needed]
CVE-2014-123456 [6 digits if needed]

http://cve.mitre.org , or come see our booth (242)

VDBs and Time-Delayed Publication

- "There were $num vulns in $year" should make you run! There were between 4,842 and 10,896 vulnerabilities in 2006 depending on the VDB you use.
- Worse… the # of vulns in 2012 changes over time as older disclosures are continually discovered and added:



(# OSVDB IDs)

CVE's "All Publicly Known Vulnerabilities" (Ten Years Later)

- Intentional exclusion
  - Site-specific (SaaS, Cloud, etc.) *
    - How to even identify these?
  - Vulns in malware ****
- Preferred (but not on sources/products list)
  - Mobile, voting machines, remote-control coffee makers, alarm clocks with built-in microphones, software that disables cars, SCADA **
- If/when we get to it
  - Joe's PHPBasketWeaving application
  - Curaguay's most popular IM application with 12 English speakers and 1.2M Curaguayan
  - Vulns from before 1999 ***

* OSF/Cloutage covers
** OSVDB covers
*** OSVDB covers
**** OSVDB covers

VDB Abstraction Bias

- Remember our ~~made up~~ *crafted* abstraction bias term? Means externally, some VDBs are basically worthless for generating meaningful stats.
- Almost no one gives criteria for a vulnerability or explains their abstraction. Secunia even disclaims their stats are not ideal (2011 yearly, page 6).
- Secunia has 28 advisories for 1 vuln (CVE-2013-1493)
- IBM 31541 = 1 entry for oracle CPU (30 different CVE).
- OSVDB has 1 entry per 1 vuln as best they can.

VDB Abstraction: 1 to 5 Entries?

CVE-1: SQL injection in version 1.x through login.php and order.php.

CVE-2: SQL injection in version 2.x through admin.php.

CVE-3: XSS in version 2.x through login.php and search.php.

ISS and Bugtraq ID

1: Mult. SQL injection in 1.x and 2.x

2: XSS in 2.x

Secunia, ISS, and Bugtraq ID

1: SQL injection and XSS in 1.x and 2.x

OSVDB

1: SQL injection in login.php

2: SQL injection in order.php

3: SQL injection in admin.php

4: XSS in login.php

5: XSS in search.php

Somebody somewhere, probably

1: login.php

2: order.php

3: admin.php

4: search.php

Abstraction Bias: Duplication Dilemma

- Unique vs. duplicate IDs
  - CVE / BID – Flag REJECTED or RETIRED
  - ISS / OSVDB – Delete outright.
- ISS and Secunia may use multiple IDs, by design, for the same issue
  - "patch action" / vulnerability / multi-vendor
- CVE has ~ 535 REJECTED entries,

  BID has ~ 550 RETIRED
- Do stats consider this? (No)

Abstraction Thoughts

- You can't reliably do trend analysis if the source changes its abstraction or coverage during the period of analysis
- IBM X-Force used to do 1 ID per Microsoft Issue:

Search for CVE numbers that contain: 2002-1182 Go

CVE	XFID	Product Coverage
CVE-2002-1182	10590	iis-ms02062-patch MS02-062

- Now, they do 2 IDs per. One for vuln, one for missing patch (to support IBM products):

Search for CVE numbers that contain: 2006-5579 Go

CVE	XFID	Product Coverage
CVE-2013-0077	81682 81683	MPEG2_DirectShow_Decompression_Abuse win-ms13kb2780091-update

Our Future

- Vulns are gonna get weirder
- Harder to measure and quantify
- Long, complex chaining of low-severity issues
- VDB's abstraction/selection/publication bias is going to be tested

Evolution bitchez!

VDB Bias is the Foundation for Bad Stats

Bad Statistics (Tunnel Vision)

(Pro Tip)

Abstraction Bias or “Counting ID’s”

Counting Vulnerabilities *

* With the exception of OSVDB who largely tries to assign one ID per root cause vulnerability. **
** And even they have to make simplifying assumptions when there’s no good data.

Survey of Past Studies – Examples of Bias

- NSS Labs, “Vulnerability Threat Trends” (2013)
- Sourcefire, “25 Years of Security Vulns” (2012)
- SC Magazine, “The ghosts of Microsoft” (2012)
- Qualys / Elinor Mills, “Firefox, Adobe top buggiest-software list” (2009)
- Gunter Ollman, "Top-10 Vulnerability Discoverers of All Time“ (2008)
- IBM Top Vuln Vendors (2007)

Good statistics are as rare as me!

Selection Bias

"The most popular applications often have the most vulnerabilities that criminals can exploit." – Harry Sverdlove, CTO of Bit9, 2010

1% of vendors account for 31% of vulns - NSS Labs, 2013

"Industry control systems (ICS/SCADA) saw more than six fold increase in vulnerabilities from 2010 to 2012" – NSS Labs, 2013

- Researcher selection bias (choosing popular software)
- VDB selection/publication bias (covering popular software)
- Vendor publication bias (being popular software)

"The number of [CVE entries] grew to 5,225 in 2012, an increase of 26 percent year-over-year" – Robert Lemos on NSS Labs report

- VDB selection bias (increased productivity)

Measurement Bias: Confusing the Units of Measurement

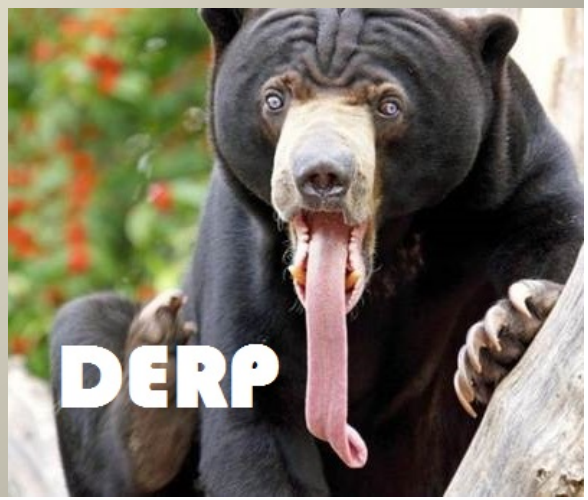
- Vulnerabilities vs. CVEs:

"The number of vulnerabilities in Adobe Reader rose [based on NVD] … while those in Microsoft Office dropped"– Elinor Mills (note: bad searches did not find all relevant products. If you do not use the VDB correctly…)

"As of January 2013 the NVD listed 53,489 vulnerabilities affecting 20,821 software products from 12,062 different software vendors." – NSS/Frei (Ph.D.)

- Vulnerabilities vs. Advisories:

"Based on my count, there were 83 vulnerabilities announced by Microsoft [in 2012]" – Alex Horan, CORE Security



Abstraction Bias: The Invalids

- Remember CVE has ~ 535 REJECTED entries, and BID has ~ 550 RETIRED?
- If somebody searches CVE but doesn't filter REJECTED, the stats will be very wrong.
- Those are just duplicates or not-a-vulnerability. What about CVE's DISPUTED?
- OSVDB tracks invalids:
  - "Myth/Fake" = 431
  - "Not a Vuln" = 76

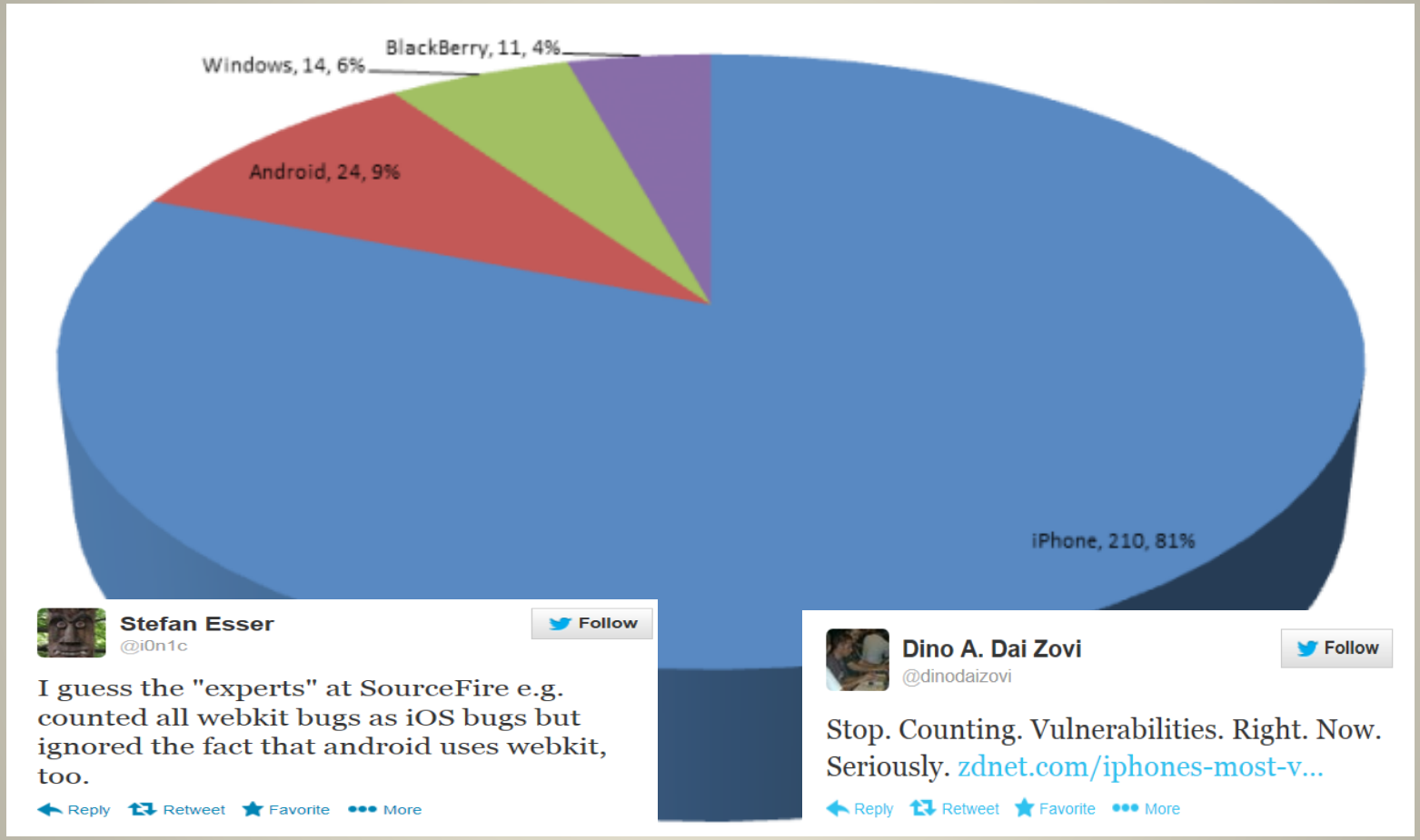
I'm not an invalid you asshole.

Windows vs. Linux

- "OS_A is more secure than OS_B!" should make you run away, fast!
- Vendor publication bias is rampant
  - It's not fair to compare one vendor who publishes everything with another vendor who only publishes most severe issues found only by external research
  - Consider selection bias of what products constitute "Windows" vs. "Linux"
  - Windows is one vendor, Linux is dozens.

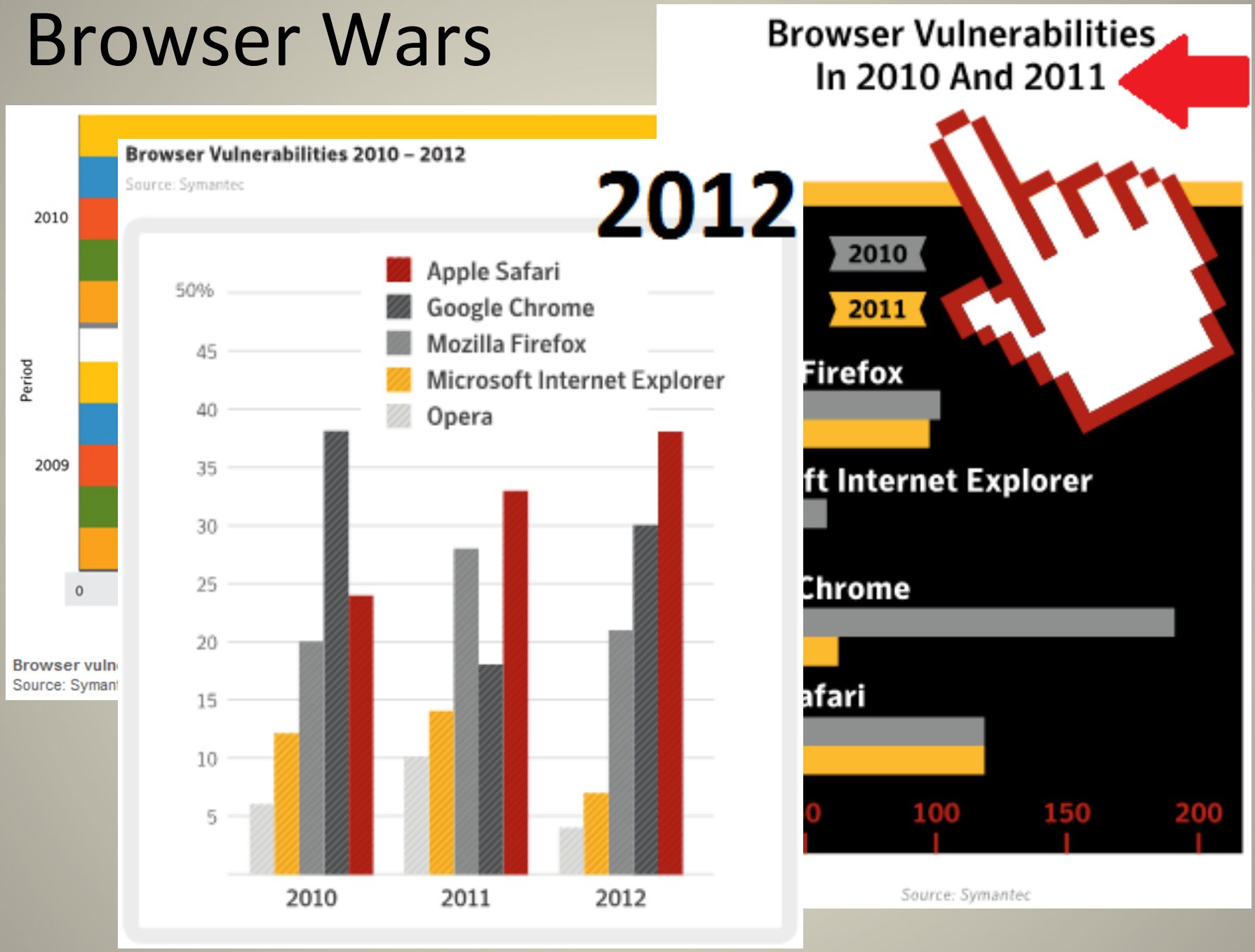
I want more web browser comparisons!

Mobile: We Don't Know Where to Begin…



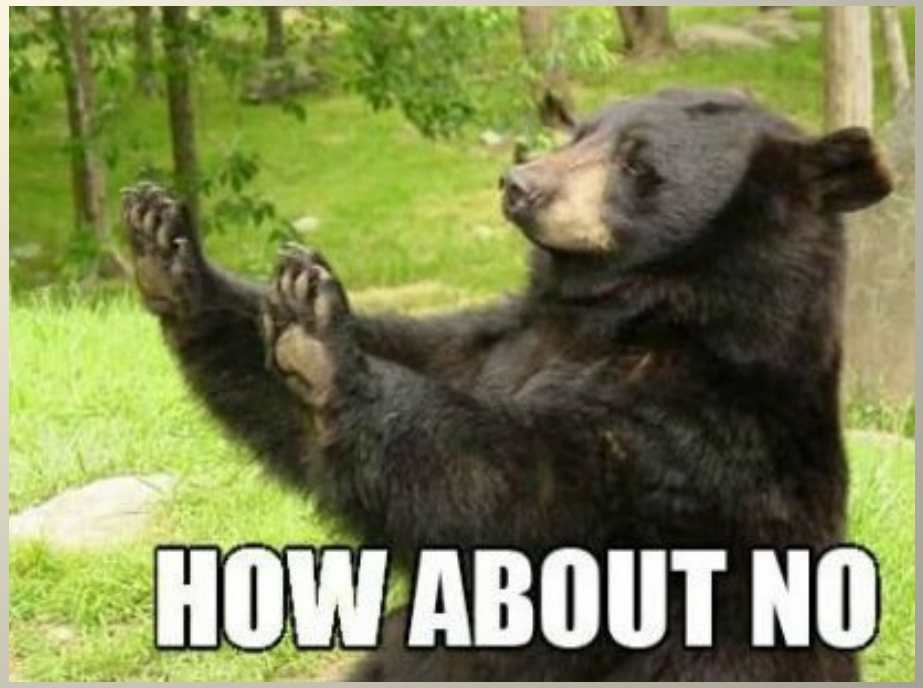
- The only take-away here? "Show your methodology"! Because this chart screams of several types of bias and no way to reproduce the findings.

Browser Wars



Product Stats: Just Don't

- … ESPECIALLY browsers!
- Why? Glad you asked!
  - Caveats. Entirely too many.
  - Unknowns. Entirely too many.
- What do we know? Those two things significantly impact stats.



Product Stats: Just Don't

- Chrome fixes WebKit vulns silently to not 0-day Apple.
  - Result: wouldn't see that vuln count as Chrome, likely see it for Apple
- WebKit → Safari vs. Chrome = Dupes may affect numbers
- How many vulns in Chrome are Chrome-specific code vs how many were WebKit or other components?
- 163 Chrome-specific vulns in 2012
  - Many in the PDF viewer, which is not their code. (From Foxit, or used to be?)
- Chrome/Firefox discloses internal/external, Safari/ MSIE discloses external only
- Mozilla memory corruptions rarely fully diagnosed

Who Discovered The Most Vulns?

- VDB selection bias: sources & coverage dictate
- Even with a great data set, so many caveats…
- 2008 Top Researchers, by Ollmann / X-Force:

Rank	Name/Handle	Total
1	*Luigi Auriemma*	612
2	*r0t*	554
3	*rgod*	357

- r0t had 935 through end of 2008 per OSVDB
- Luigi had 615 through end of 2008 per OSVDB

Prolific Researchers: Jan 2012 – Now (#OSVDB)

R	Researcher	#	Caveats
1	Mateusz "j00ru" Jurczyk	411	DoS/Code Exec vs Stability?
2	Gynvael Coldwind	345	Adobe/MS/Chrome/FFMpeg [Solid Research]
3	Benjamin Kunz Mejri	240	How many myth/fake or not-a-vuln?
4	High-Tech Bridge SA	237	Team of researchers, day job to disclose
5	Suman Jana	192	Few common vulns found in a ton of software
6	Janek Vind "waraxe"	177	22 disclosures + abstraction
7	Vitaly Shmatikov	170	Research partner with Suman Jana
8	Abhishek Arya (Inferno)	142	Mozilla/Webkit/FFMpeg/Chrome [Solid Research]
9	Gjoko Krstic	135	Good variety, some abstraction bias

- 300+ can be 3% or more of ALL disclosures in a year
- A single researcher can cover 50% or more of all disclosures for a particular product or product class

The Don't Do of Stats

- Don't do pie or bar charts that compare products based on vuln total or percent of vulns
- This varies too widely between products.
  - Researcher/Selection
  - Vendor/Publication
  - VDB/Measurement (to a lesser degree)

From Bad to Good

Good Commentary

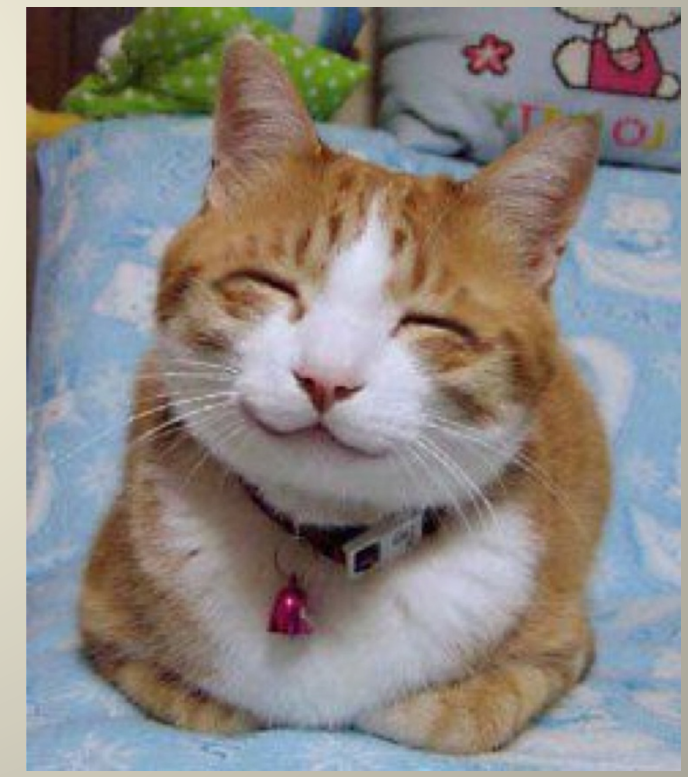
"… vulnerability count reports [like the report from Bit9] seek to measure a complex, multi-faceted problem from a single dimension. It's a bit like trying gauge the relative quality of different Swiss cheese brands by comparing the number of holes in each: The result offers almost no insight into the quality and integrity of the overall product, and in all likelihood leads to erroneous and - even humorous - conclusions." - Brian Krebs

I know, I can't believe it either!

Good: Normalize by 'Risk' not Vuln Counts

- "Risk" is… um… relative to YOU (thinking required)
- "Windows of Vulnerability" Article (Brian Krebs)
  - Publishes data? YES

"For a total 284 days in 2006 … exploit code for known, unpatched critical flaws in pre-IE7 versions of the browser was publicly available on the Internet. Likewise, there were at least 98 days last year in which no software fixes from Microsoft were available to fix IE flaws that criminals were actively using… In contrast… Firefox … experienced a single period lasting just nine days last year in which exploit code for a serious security hole was [widely public before Mozilla had a patch]." – Brian Krebs



Good: Recognition of Vendor Publication Bias

- It is easy to clearly disclaim vendor publication bias, yet rare to see.

"… the high number of Firefox vulnerabilities doesn't necessarily mean the Web browser actually has the most bugs; it just means it has the most reported holes… proprietary software makers … typically only publicly disclose holes that were found by researchers outside the company, and not ones discovered internally." – Elinor Mills

Good: Disclose Your Methods

- Simply citing the VDB you use is not disclosing your method!
- Bit9 report listed specific selection criteria: end-user applications, NVD entries published between Jan and Oct 2010, CVSS 7-10 based on NVD.
  - Also bad (selection bias): Bit9 protects desktop apps
  - Also bad (VDB bias): NVD has coverage gaps

Wait, that's more bad than good?!

Good: Show Your Selection Bias

"… it's entirely possible that some vulnerabilities may be missed because they were disclosed on non-public lists or couldn't be verified as actually being a real vulnerability." – Gunter Ollman

- "entirely possible" means it <u>may</u> be true. There is no 'may', there is '<u>absolutely</u> happens' or 'unknown'.
- Yet CVSS operates off "<u>may</u> be true". Woe is us!

Good: Show Your Abstraction Bias

"Multiple vulnerabilities affecting the same product, the same version, and through the same vector, but with different parameters will be counted as a single vulnerability by X-Force (since they will require the same remediation and/or protection)." – Gunter Ollman

- That's the good, but as noted, X-Force changed their abstraction method once, maybe twice.
- You did notice that…
- Now what?

Now, put your armor on to protect you from that Jericho bastard.

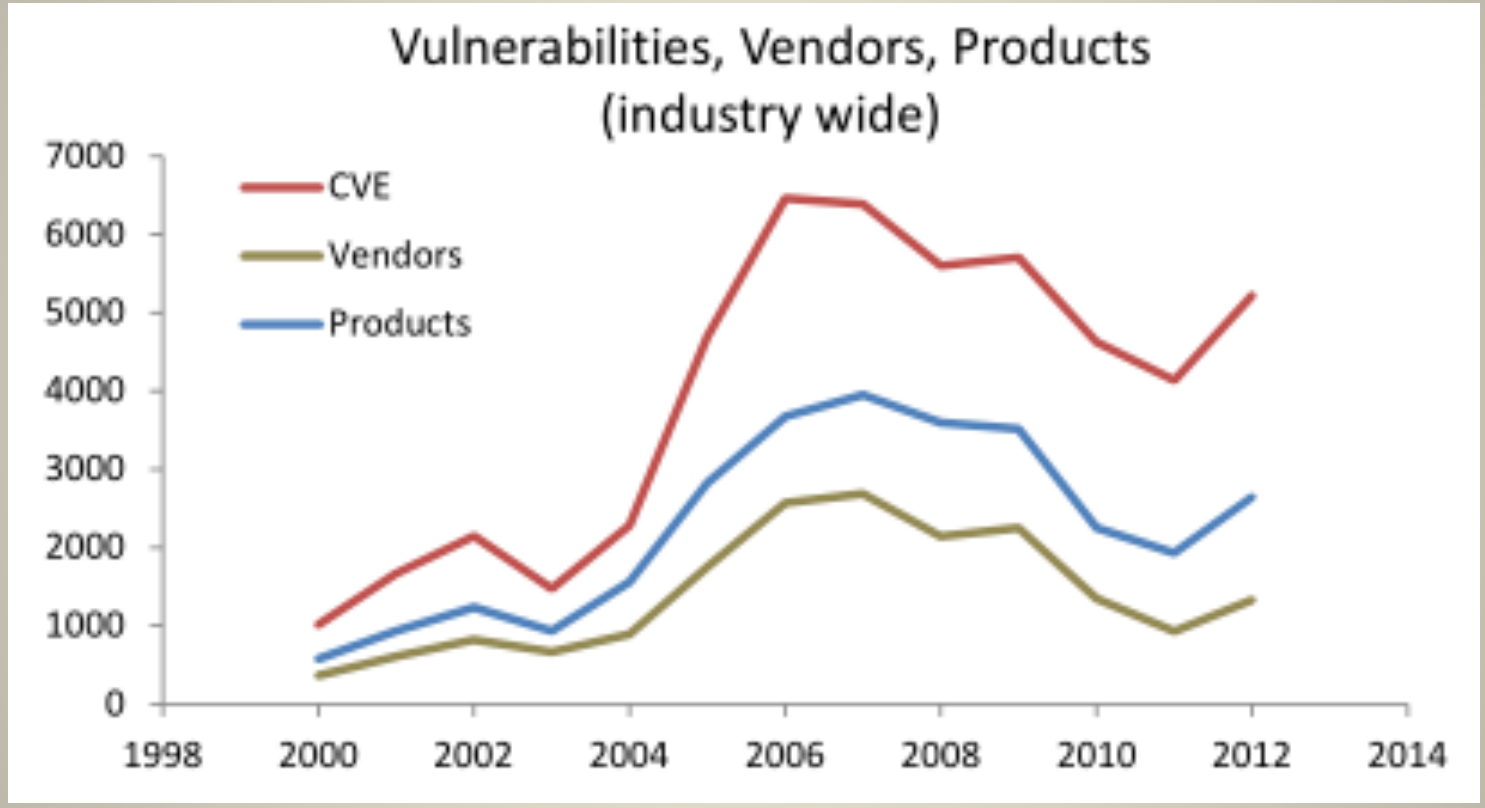
Good: Units of Measurement

"… the number of vulnerabilities [was] an increase of 26 percent year-over-year, as counted by their [CVE] identifiers." – Robert Lemos on NSS Labs report

- Definitely good, but do you also specifically outline why counting by a CVE is bad? If not, laymen will assume CVE == vuln.
- Vuln stats are sorcery**. Explain it to them gently.

Good Methods, Bad Data – Part 1

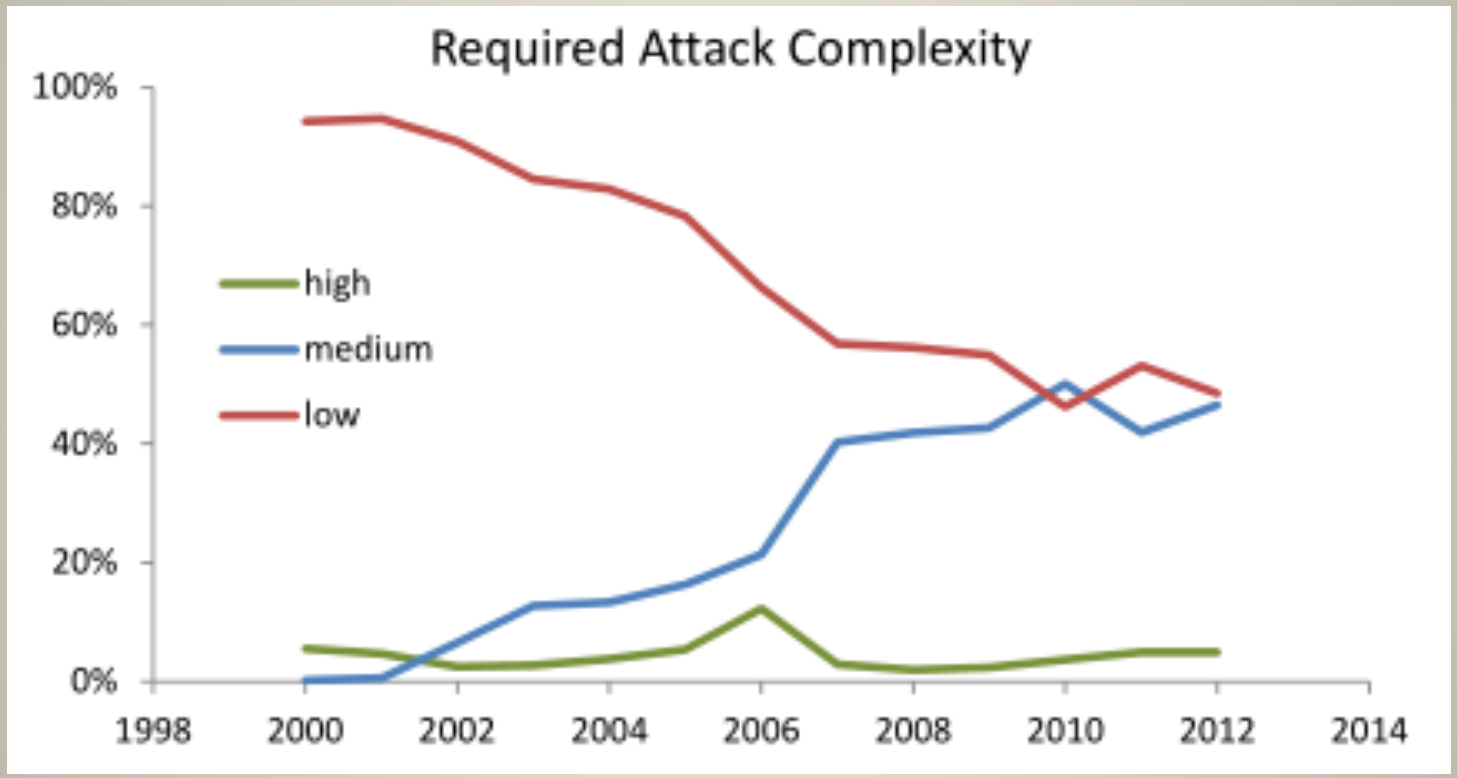
- Good: NSS Labs visualization of vendors/vulns:



- Bad: Major vendors part of larger ecosystem, not reflected in CVE data (better coverage of “Joe Schmoe” vendors in 2006 than 2013)

Good Methods, Bad Data – Part 2

- Good: NSS Labs attack complexity graph:



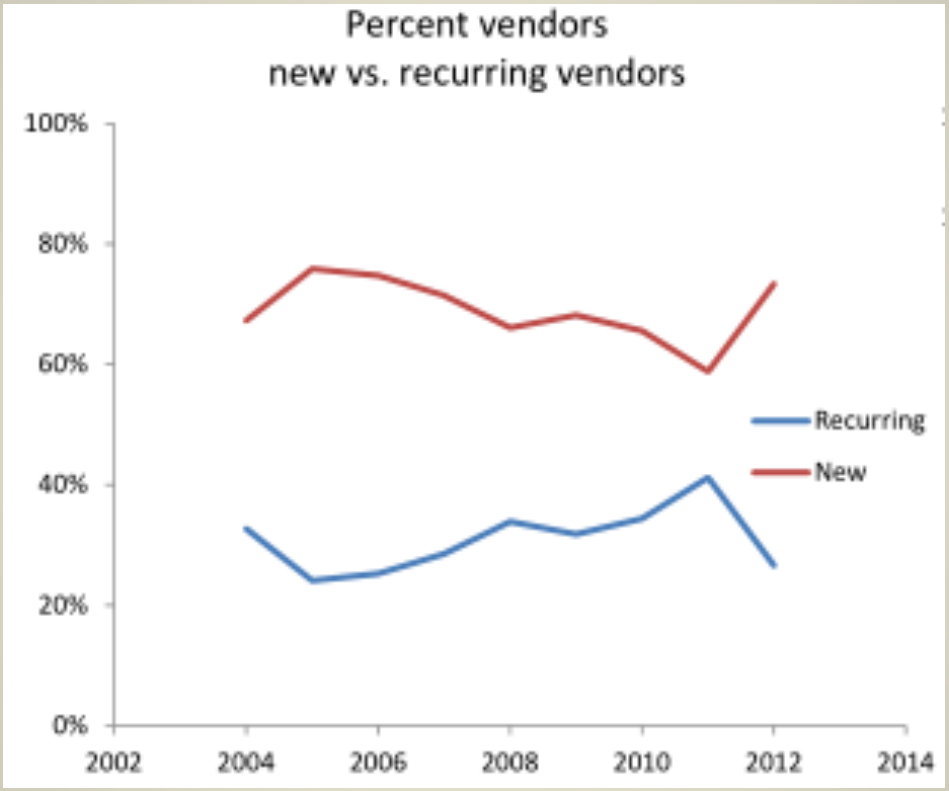
- Bad: CVSS 'Access Complexity' significantly lacks granularity. Severity-based publication bias.

Good: Using CVSS ~~Correctly~~ Less Wrongly

- The CVSS vectors are a ~~gold~~ silver mine of better, more granular information
- Still not granular enough for experts such as this audience (you know you're the minority, right?)
- CVSS authentication requirements, or remote vs. local
- Look at distribution of CVSS scores, not raw counts

Good Methods, Bad Data – Part 3

- Good: NSS Labs new vs. recurring vendors:



- Bad: partially due to increased CVE coverage? More vendor types? (if CVE doesn't know for sure, maybe you don't know for sure either?)

Telling Good from Bad

- Were sources cited?
- Was source's coverage consistent?
- If multiple sources used, are they consistent in their selection, publication, and abstraction?
  - (Answer: NO)
- Was the methodology for collecting and interpreting data documented?
- Were the units of measurement correctly described?
- Important: Were ALL of the above done?

Departing Observations…

- We are in the "leeches" stage of infosec, in multiple senses of the word.
- If you can't measure it, you can't manage it.
  - For now, "we complain about it, so you can help fix it"
- Researchers, Vendors, VDBs all have bias, that may conflict with each other, this wreaks havoc on stats.
- Stats are currently about the "what"; we need more of the "how" and "why"

For More Exposure…

- VIM
- OSVDB blog
- Critical thought
- OSS-Sec (if a masochist)
- CVE Editor Commentary
- Any good beginner statistics book…

Questions?

OLD, UNUSED, OR BACKGROUND SLIDES

Still-Active Ideas

- can also do one for the escalation of xss/sqli to overflow dos to overflow to mem corruption etc. i like those ideas

- Should duplicate vs. unique IDs be considered a part of abstraction bias? Applies to VDBs and some vendors especially. [Only if we can prove that VDBs have dupes and don't fix them. CVE fixed in a different manner than most, and may introduce bias.]

OSVDB stats: most popular researchers

- shows how the most influential on stats are not necessarily
- well-known
- distribution of vuln types?
- distribution of severity
- show how some researchers start with simple vuln types, then
- expand; product types, too
- the X most popular researchers, and do a line plot of their production over recent years. This would show how a researcher's numbers rise

Multiple Types of "Vulnerability" IDs: The ABCs

- Advisory ID
  - MS12-067 (Microsoft), SA12345 (Secunia), …
  - No ID: Oracle, Cisco, …
  - HP (multiple IDs)
- Bug ID (often "Vulnerability")
  - CERT-VU, JVN, Cisco Bug ID, OSVDB, …
  - Rarely used by researchers
- Coordination ID
  - CVE-xxxx-yyy
- Many things have more than one ID
  - cars, computers, books, humans, …
- Each ID type serves different purposes and audiences
- One ID type can be used (poorly) for a different type of thing

(counting is only from publisher's perspective)

(counting must be usable by multiple perspectives)

Predicting Popular Vulnerability Classes

- A class may become popular if it has all of these:
  - Bad consequences
    - Remote code execution, data compromise, security bypass
  - Easy to find
  - Easy to write exploit code
  - Has had a white paper or two written about it
  - Has hit very popular software
- Past examples: buffer overflows, format strings, SQL injection, PHP file inclusion, XSS, CSRF

Generally there seems to be at least a 2-year lag time between first discovery and rampant exploitation. Exception: format strings.

Confirmation bias

- Omitting this notion because it's a cognitive bias, not statistical
- "tendency of people to favor information that confirms their beliefs or hypotheses"
  - "Unbreakable" Oracle and David Litchfield
  - "[X] is more secure than [Y]"!!

Vendor Disclosure Practices

	Red Hat	Mo zilla	Google	MS	Juniper	HP	Oracle	Adobe?	Cisco	Apple
Own discoveries	Y	Y	Y	N	No/ Yes?		Y			
Low-severity	Y						Y			
Tech details	Y	Y	Meh	Meh	Meh	N	N	Meh	Y	Y
EZ public adv access	Y	Y	-	Y	**No/ Yes**	Y	Y	Y	Y	Y
Public 0day comment				Yes			No	Yes		No
Cross-refs	Y		No	No	No	No	No	No	No	No
Bug-level access	Yes	Ltd		Priv ?	No	No	No	No	Ltd	No

Counting

Counting/ Severity

Affects counting

IBM is not EZ

Affects accuracy

Accuracy/ dupes

Accuracy

“Meta-analysis”

- “methods focused on contrasting and combining results from different studies, in the hope of identifying patterns among study results, sources of disagreement among those results, or other interesting relationships that may come to light in the context of multiple studies”

Steps in Meta-Analysis

- (from wikipedia)
- …
- Selection of studies … Based on quality criteria, e.g. the requirement of randomization and blinding in a clinical trial

Realization?

- Maintaining VDBs is like performing a meta-analysis on many sources, each with their own biases
- Using a VDB for large-scale statistics is akin to a meta-meta-analysis

Sampling Practices in VDBs

- VDBs typically try to do large samples
  - Typically, the intended population is "everything we can find"
  - All of us, collectively, still don't know everything
- Meta-analysis is comparing or combining samples from different entities
  - HA ha ha ha ha

Participation bias

- "the results of elections, studies, polls, etc. become non-representative because the participants disproportionately possess certain traits which affect the outcome"
  - Are major vendors "victims" of this?
- Individual researcher productivity

Epidemiology (Disease Research) Versus Vulnerability Research

- Blah blah blah, basic intro?
- Ultimate goal: improve "health" of the Internet
  - Vulns/Attacks are the disease
- A "sample" is a selection of some subset of objects from a population
  - E.g., humans
- An "object" is one vulnerability (however we count those)
  - These are not as discrete as "people" or "rats"
- A "population" is a set of vulnerabilities
- A sample is a collection of vulns by a single entity over time, e.g. a vendor, researcher, or VDB

Funding bias

- “an observed tendency of the conclusion of a scientific research study to support the interests of the study's financial sponsor” (Wikipedia)
- ** Not quite matching these examples **
- Researchers: product selection can be the direct result of their employment
- VDBs: whether profit or non-profit, must address its customers’ requirements, which hurts usability by other groups

Systematic errors

- *I don't like the sound of this…*
- "imperfect calibration of measurement instruments (zero error)"
  - If a thermometer is always off-by-one, then this can be adjusted
- "changes in the environment which interfere with the measurement process"

- Actually, this can be accounted for (potentially) if known, more so than "random errors"
- Random errors can be hard to detect
- Infosec wants good randomness, so maybe this is by design? ;-)

Well-Known Bias Problems in Vuln Stats

- Selection bias (which prods/vuln-types researchers look for)
- Confirmation bias (vuln classification)
- Reporting bias (vendors/VDBs)
- Researchers low-severity / "embarrassing" XSS types

CVE Abstraction ("Counting") Versus Other Approaches

- CVE's level of abstraction has evolved to be IN THE MIDDLE
  - Maximizes utility to many communities
- The content decisions rely on information that is usually stable, and often published early
- Can be difficult to "count" correctly and consistently
- Still affected by what information is available at the time of assignment
- Less flexibility to change our minds after CVE-IDs are publicly used

Remove?

- Known unknown (multiple unspecified)
- Unknown unknowns
  - Vendor discovered, fixed, no advisory
  - Undisclosed 0day
  - Disclosed, but lost in history

Exclusion Bias

- A type of sampling bias: “exclusion of particular groups from the sample”
  - Some researchers won’t target unpopular software or “lame” vulns
  - Some VDBs won’t include vulns without a patch (?!), some from unreliable sources, some due to lack of time

Examples of Bias in Vuln Research

	Researchers	Vendors	VDBs
Selection bias	Choose particular products or vuln types to research	Conduct internal research based on internal priorities; work with external researchers	Monitor certain disclosure sources
Publication bias	Might publish only for high-profile products; avoid low-risk and "lame" vuln types	Only publish patched, high-severity issues for currently supported products & versions	Only publish "verified" issues of a certain severity for "supported" products
Abstraction bias	Release many advisories for one core issue, boosting counts	Combine many vulns into the same advisory for sysadmin convenience	Use the level that is best for the intended audience
Measure-ment bias	Over-estimates severity, or does not validate findings	Under-estimates severity or uses generic vuln terms	Misinterprets external disclosures

Grep-and-Gripe 2: Larry Cashdollar*

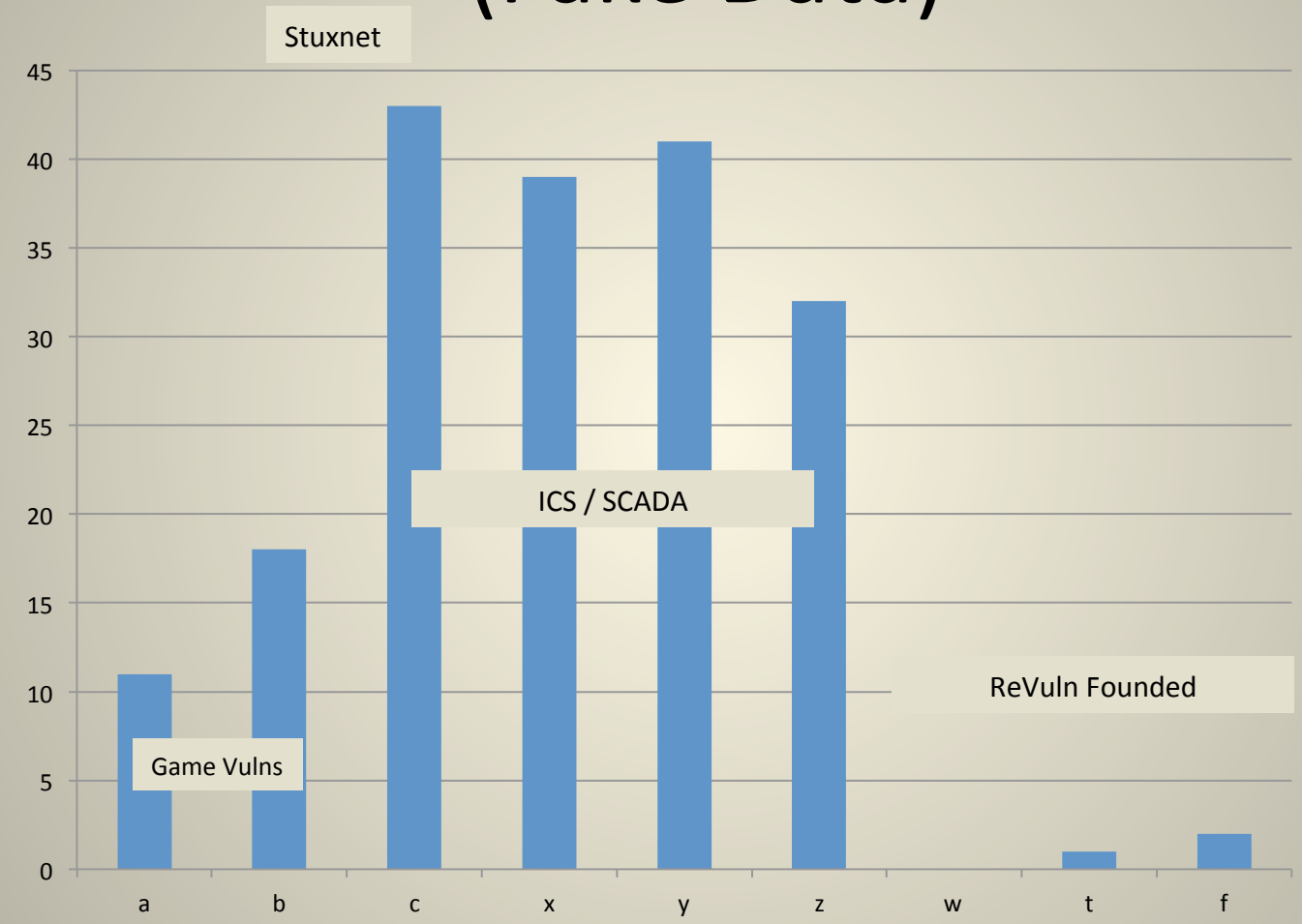
* That's his real last name. He swears it!

- Grep-and-gripe
- Old-school symbolic links and context-dependent OS command injection
- Those are dead, right?
- Enter Ruby Gems

"ruby"

"ruby gem"

Luigi Auriemma – Published Vulns (Fake Data)



EXTRANEOUS NOW?

- Accuracy problems by researchers?
  - Grep-and-gripe!
  - A back-tick is always SQL injection!
  - That crash surely shows signs of code execution!

CVSS

Not real happy. CVSS is a single element of VDBs and stats. I know it is important given the weight it receives. I don't see a good point to do this section to keep the flow though.

Bias and CVSS

- Measurement bias
  - Emphasis only on impact to system
  - Vagueness/inconsistency in application by different people
- … which introduces selection bias
  - E.g. “only CVSS > 6.0”
- CVSSv3 under development
- ~~… And that’s all we gotta say about that.~~

Publication Bias Defined

- "a bias with regard to what is likely to be published, among what is available to be published" (Wikipedia)
- "the practice of selectively publishing trial results that serve an agenda" (Mercola.com)
  - "half of all clinical trials ever completed on [current] medical treatments have never been published in the medical literature"

Reporting Bias

- "a tendency to under-report unexpected or undesirable experimental results" by subjects
- Nobody discloses their failure to find vulns
  - a.k.a File Drawer Effect?
- Social desirability bias – “tendency of respondents to answer questions in a manner that will be viewed favorably by others”

Reporting Bias: Examples

- Researchers - Not every researcher discloses, and not every researcher discloses everything they found. Legal threats stifle disclosures
- Vendors – Typically do not disclose their internal findings.
- VDBs – Might not report vulns they accidentally discover during research

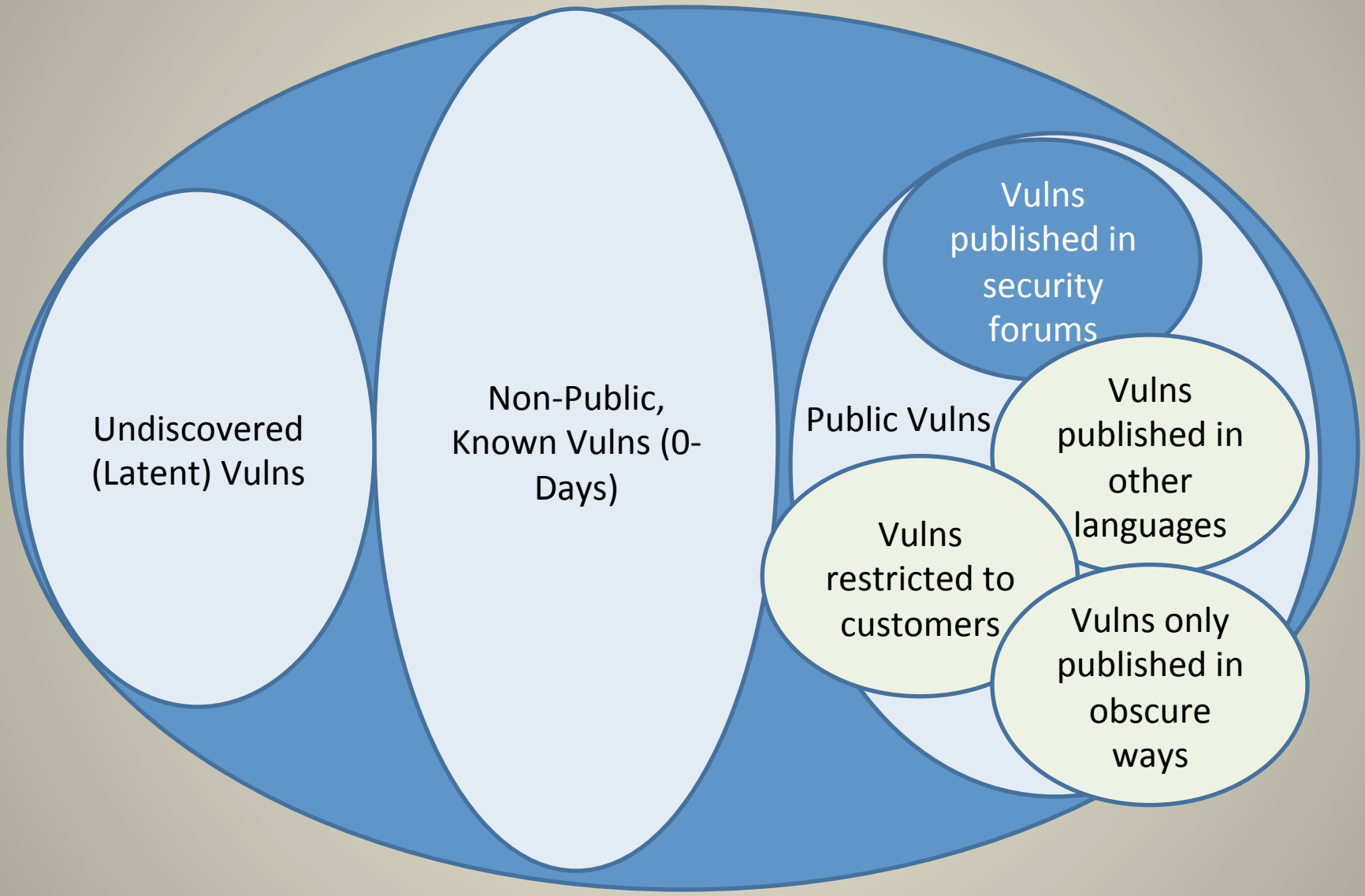
Vendor Practices (In Progress)

	Yes		No
Publication Bias			
Public advisory access	Red Hat, Mozilla, MS, HP, Oracle, Adobe, Cisco, Apple		Google (no advisories), Juniper (no publication until 2013), Linux kernel
Own discoveries	Red Hat, Mozilla, Google, Oracle		Microsoft, ??? Linux kernel, Juniper didn't publish at all until 2013
Low-severity	Red Hat, Oracle, ???		Microsoft, ???
Measurement Bias			
Tech details	Red Hat, Mozilla, Apple. Meh: Google, MS, Juniper, Adobe		HP, Oracle
Public 0-day comment	Adobe, Microsoft, Red Hat.		Oracle, Apple
Cross-refs	Red Hat. Unknown: Google, Juniper, Mozilla		HP, Oracle, Adobe, Cisco, Apple
Bug-ID access	Red Hat	Ltd: Mozilla, Cisco, Google	HP, Oracle, Adobe, Apple

VDB Biases

- for each, will give examples for various vuln DBs
- Analytical effort: how much effort does the VDB analyst put into producing the VDB entry?
  - CVE: show increase in desc size
  - Amount of details
- Description, other fields
- Vuln classification
- Custom channel for new vulns?

The World of Vulnerabilities



FACT: No VDB knows how many vulns were disclosed.

Fuzzmarking (replacement in studies)

- Kaminsky, Cecchetti, Eddington (2011)
- They used the same fuzzer against Windows Office and OpenOffice from 2003, 2007, and 2010
  - Selection bias: use same environments and time frames
  - Publication bias: ???
  - Abstraction: ???
  - Measurement bias: use same tools