SEVENTH FRAMEWORK PROGRAMME

MASTER
Managing and Auditing
Security and Trust for sERvices

Università degli Studi di Trento

secure CHANGE

Which are the right sources for vulnerability studies? A case study on Firefox

Fabio Massacci
Viet Hung Nguyen
University of Trento

MetriCon @ USENIC Security - 2010

8/10/2010

Know thy speaker

- **Phd in Formal Method/Logic for security**
  - **But I hacked a major conf web site and could assign myself reviews so I become…**
- **Professor in Computer Security**
  - **Co-founded Quality-of-Protection/Metrisec workshop**
  - **Compliance, security metrics, smart card, mobile security**
- **Deputy rector for ICT services and procurements for 7 years at my university**
  - **70+ IT staff, 7+MEuros/year in contracts**
    - I was the “so what?” guy
    - and could ditch a security project budget with a stroke of a pen

Lots of Metrics on Vulnerabilities Discovery, Evolution…

- “Handwaving Guru” Models
  - Anderson, Littlewood and Strigini, etc.
  - Most Models of Economics of Security
- “Out-of-the-hat” metrics
  - Manadhata, Wing et al (Attack surfaces)
- “Line-through-asteroids” Experimental Models
  - Ozment and Schechter, Alhazmi and Malaiya, Frei et al.
- Simulation-based Epidemiology Models (eg virus)
  - Chakrabarti et al.
- Machine-Learning Predicting Faulty Components
  - Neuhaus et al. Gegick et al, Chowdhury & Zulkernine, etc.

Basic Ideas (of sound works)

1. Measure #Characteristics for Sw
   - Version/Component 1 … n-1
2. Measure #Vulnerabilities for Sw
   - Version/Component 1 … n-1
3. Find some correlation
4. Use correlation to predict #Vulnerability
   - On Version/Component n

- Apparently actionable
  - IF Predicted Vul n>threshold THEN more testing effort, put behind firewall etc. etc.

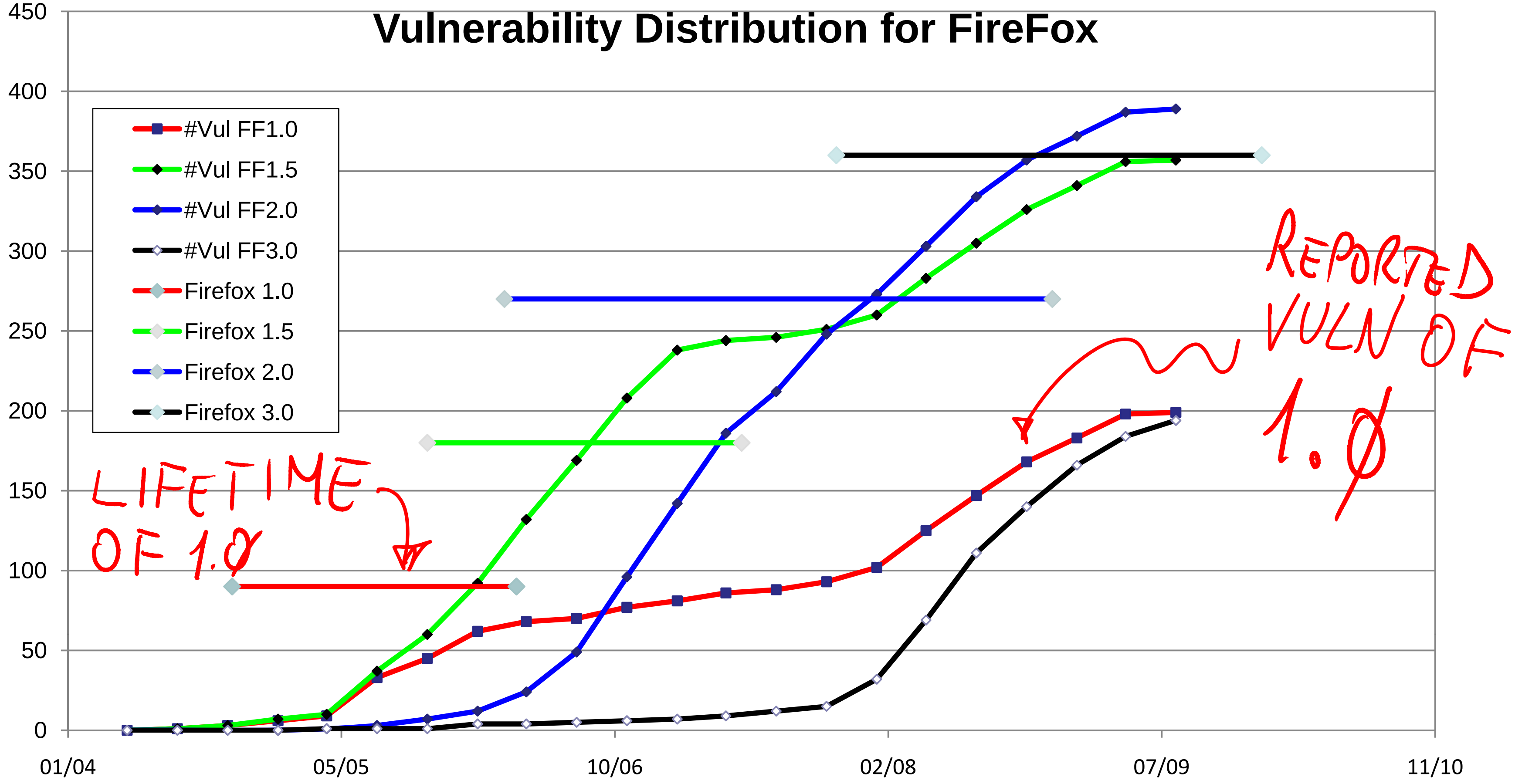
How to Measure Vulnerabilities?

- The obvious one
  - Mozilla Foundation Security Advisories DB
- The popular one
  - Common Vulnerability and Exposures DB
- The less obvious ones
  - National Vulnerability DB
  - Mozilla Firefox CVS (main tags)
- So we just tried to do a major experimental study

Mozilla Study

- **Integrated Code & Vulns**
  - **all* vulnerability dbs CVE, MFSA, NVD, Bugtraq**
  - **CVS Firefox 1.0→3.0**
    - 4 years of code updates
    - tracking the life of each line
  - **currently integrating 3.5-3.6**
    - Mozilla changed repository structure
- **Tried all* possible code metrics**
    - More data to apppear in Metrisec 2010 at ESEM

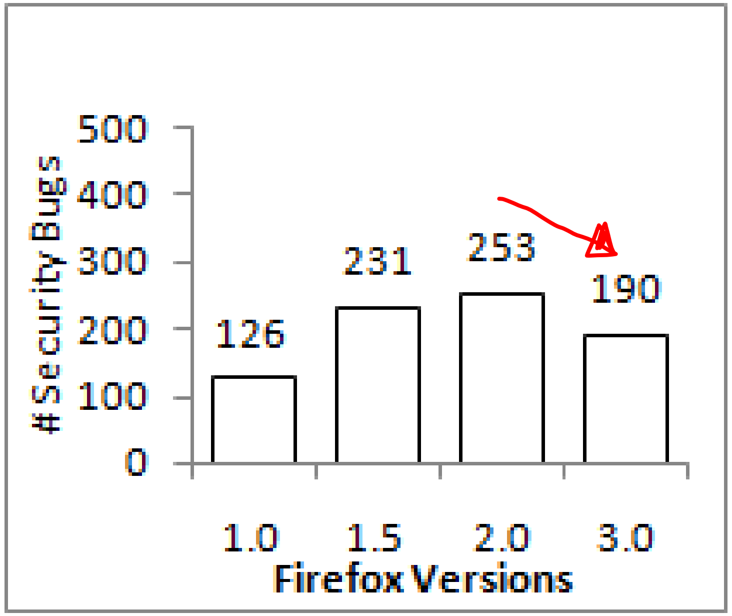
We started getting strange results…

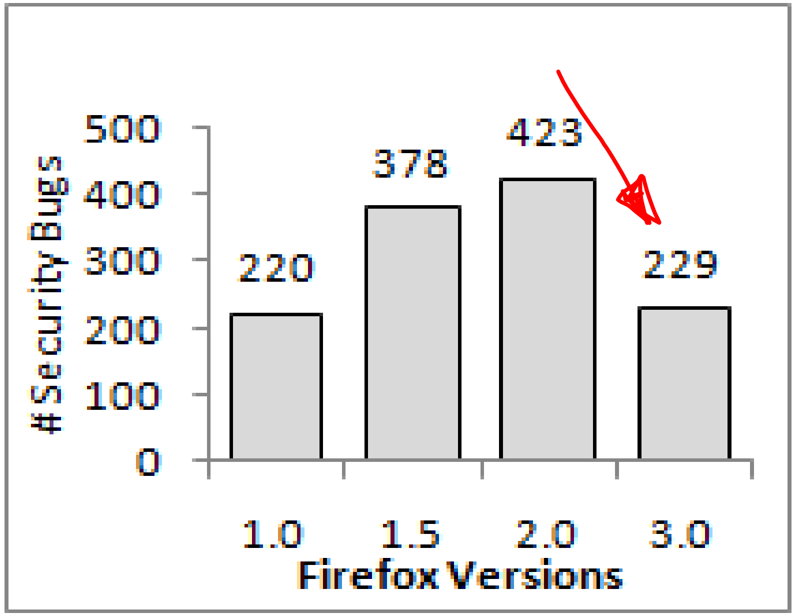


Università degli Studi di Trento

MASTER
Managing and Auditing Security and Trust for sERvices

Ooops 1: MFSA vs NVD





- for MFSA 3.0 improves 2.0 by 25%, for NVD by 46%!!!
- MFSA missed 30-40% of Vulns but NVD doesn’t tell where they are…
- MFSA fixed vulns, NVD present vulns: you can locate the former but want to predict the latter…

The Obvious Observation

- If we correlate a precise metric with an unprecise one we cannot obviously get a precise prediction

  → our (re)action will often be off the mark

- The key is how off and how often?

  1. If we are not too off, this approach works
  2. If we will "always" be off the mark maybe we need a different strategy

- Our case study suggest → (2)

The fallacy is in the word “Measure”

- “Measure” #Characteristics for Sw
  - Precise, repeatable, uniform metrics at level of components.
  - can write code that achieve target #Characts.
  - In Economics -> Micro-economic
- “Measure” #Vulnerabilities for Sw
  - Precise? Repeatable? Uniform?
  - We can’t write code with a target #Vuln
  - Only at Macroscopic Level -> Macro-Economics

MFSA – Date of infection and vaccinated individuals

- MFSA 2009-35
  - Title: Crash and remote code execution during Flash player unloading
  - Impact: Critical
  - Announced: July 21, 2009
  - Reporter: Attila Suszter
  - Products: Firefox
  - Fixed in: Firefox 3.5.1, Firefox 3.0.12
  - References to Bugzilla and CVE

DATE

- Precise (more or less), Repeatable?

CVE – The press-release of the virus

- CVE-2009-2467
  - Description
    - Mozilla Firefox before 3.0.12 and 3.5 before 3.5.1 allows remote attackers to cause a denial of service (application crash) or possibly execute arbitrary code via vectors involving a Flash object, a slow script dialog, and the unloading of the Flash plugin, which triggers attempted use of a deleted object
  - References to NVD
    - A lot of other references
- Little that can be automatically processed
- Precise? Uniform? Repeatable?

NVD I – The Health-Care Authority Notices

- **Vulnerability Summary for CVE-2009-2467**
  - **Original release date:07/22/2009 + Last revised:09/04/2009**
  - **Overview = CVE**
  - **Impact**
    - CVSS Severity (version 2.0):
    - CVSS v2 Base Score:10.0 (HIGH) (AV:N/AC:L/Au:N/C:C/I:C/A:C) (legend)
    - Impact Subscore: 10.0
    - Exploitability Subscore: 10.0
  - **CVSS Version 2 Metrics:**
    - Access Vector: Network exploitable
    - Access Complexity: Low
    - Authentication: Not required to exploit
    - Impact Type:Allows unauthorized disclosure of information; Allows unauthorized modification; Allows disruption of service
- **Lots of “opinions” that can be automatically processed**
  - **(why high? How unauth modif happens?)**
- **Uniform? Precise? Repeatable?**

NVD – II: Track of infected individuals

- **Vulnerability Summary for CVE-2009-2467**
  - **Vulnerable software and versions**
    - mozilla:firefox:2.0.0.14
    - ...
    - mozilla:firefox:1.0.8
    - …
    - mozilla:firefox:3.5
  - **84 entries of different versions of software**
- **No dates but combined with MFSA can be used to determine a vulnerability discovery metric**
- **Precise (more or less), repeatable?**
- **Notice:**
  - **vulnerability has been discovered for 3.0 (and 3.5) and is applicable to 1.0.8 but has not been discovered for 1.0.8**

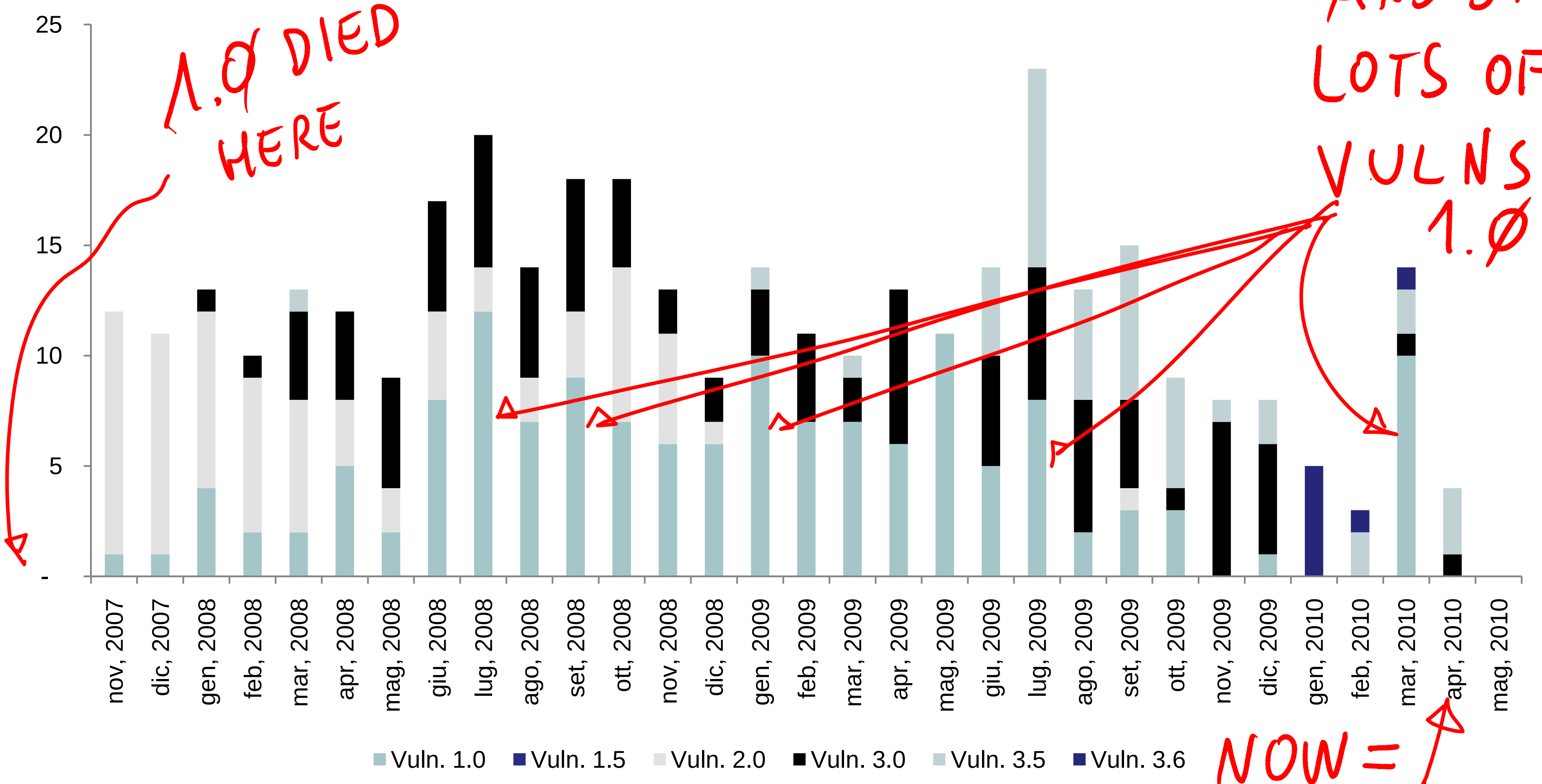
To be actionable: When Stop Measuring and Start Acting?

- “support for older versions of Firefox typically ends six months after a new major version is available”

Ver	Supp	Birth	Death
1.0	No	Nov, 2004	Apr, 2006
1.5	No	Nov, 2005	May, 2007
2.0	No	Oct, 2006	Dec, 2008
3.0	Yes	Jun, 2008	(for sec. updates)
3.5	Yes	Jun, 2009	
3.6	Yes	Jan, 2010	

- Natural Acting Pattern (for MFSA/NVD)
  - Measure 1.0 and v.5 till 2007 predict on 3.0 in 2008
  - Measure 1.0-→2.0 till 2008 predict on 3.5 in 2009
- Is this meaningful?

Oops 2: nobody can keep a good vulnerability down…



1.0 DIED HERE

AND STILL LOTS OF VULNS ON 1.0

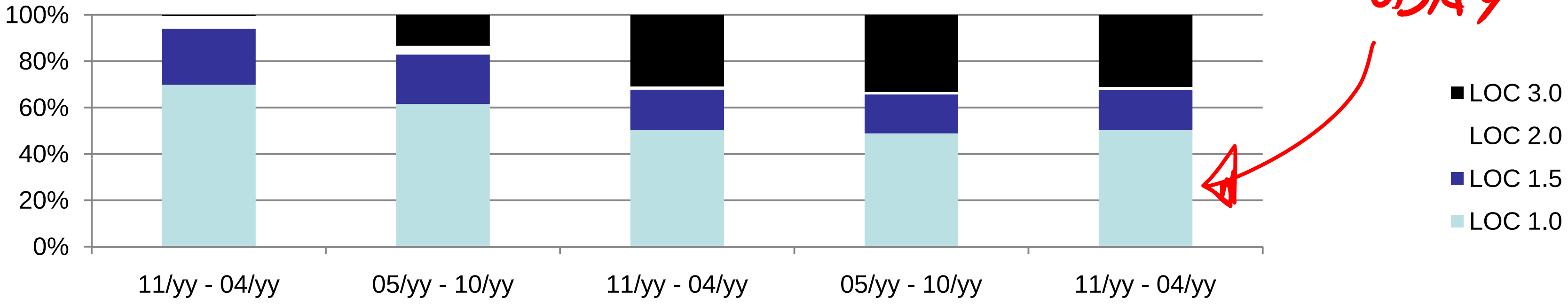
NOW = 4 YEAR AFTER!

And you can’t even claim that 1.0 is not relevant

- NetMarketShare (Jan 2010)

– 3.6	1.15%	infant
– 3.5	17.08%	adult
– 3.0	5.24%	ought to be dead
– 2.0	0.78%	… dead since 1.2yrs
– 1.5	0.10%	… dead since 2.7 yrs
– 1.0	0.03%	… dead since 3.8 yrs

NOT ONLY ISN'T DEAD BUT HAS A HUGE FRACTION OF THE CODE BASE OF TODAY



Conclusions?

- Where's the fallacy?
  - #Vulnerabilities are Macro-Economic variables you can't use them to control Micro-Economics variables (eg which sw gets double testing)
- Rather use information to change process eg
  - We can't predict well which NEW components will be vulnerable but
  - We know 20% vulns found 3yrs after release
  - We know 1-5% of legacy software always in use

→So we must have production, deployment and execution environments able to cope for that