Software Security Economics: Theory, in Practice

An Exploratory Analysis

Stephan Neuhaus<neuhaust@tik.ee.ethz.ch>
Bernhard Plattner <plattner@tik.ee.ethz.ch>

"Making the Best Use of Cybersecurity Economic Models"

- Investing in software security always has positive, but diminishing returns
- Modeled by a "increasing convex function", which is "any increasing twice continuously differentiable function"
- Statement without any qualification

Rachel Rue and Shari Lawrence Pfleeger. *Making the best use of cybersecurity economic models*. IEEE Security & Privacy, **7**:52–60, 2009.

Security

ΔS_2

ΔS_1

$\Delta S_2 < \Delta S_1$

Cumulative Investment

ΔI

Security Functions

- Increasing (Slope always positive): $df/dI > 0$
- Diminishing returns (Later slopes smaller than earlier ones): $d^2f/dI^2 < 0$
- Hang on, that's not convexity, that's *concavity*
- Not just *any* old twice continuously differentiable function will do
- Also, twice *continuous* differentiability not needed, twice differentiable suffices

What they say is not what they mean

Is what they mean actually true?

More Security Functions

- Investment *always* yields positive returns?
- Just throw money in the general direction of security and it will *never* get any worse?

- That would mean that it is impossible to choose a wrong security mechanism
- Security systems so badly implemented that *no* security system would have been better! (TSA)
- Just a *plausibility argument*, what does the *data* say?

Consequences

- Vulnerability fixing is security investment
- Stretched over time
- If same investment yields less improvement tomorrow than it does today, then vulnerability fix rates should go down

$$\text{fix rate} = \frac{\#\text{vulns fixed} \times \#\text{fixers}}{\text{unit time}}$$

Data Sets

- Mozilla (292 vulnerabilities)
- Apache httpd (66 vulnerabilities)
- Apache Tomcat (21 vulnerabilities)

Mozilla

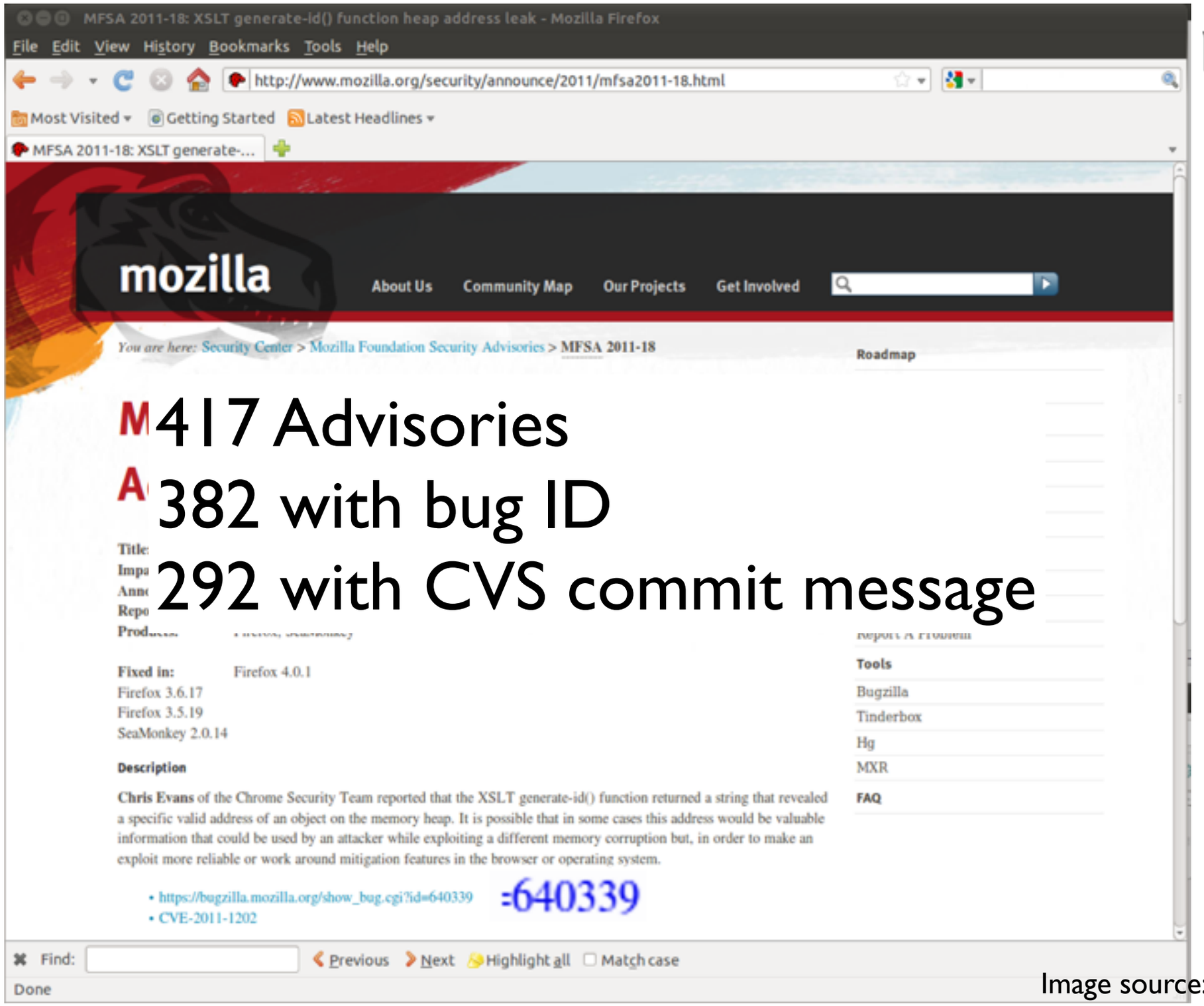


Image source: Mozilla foundation

Apache httpd

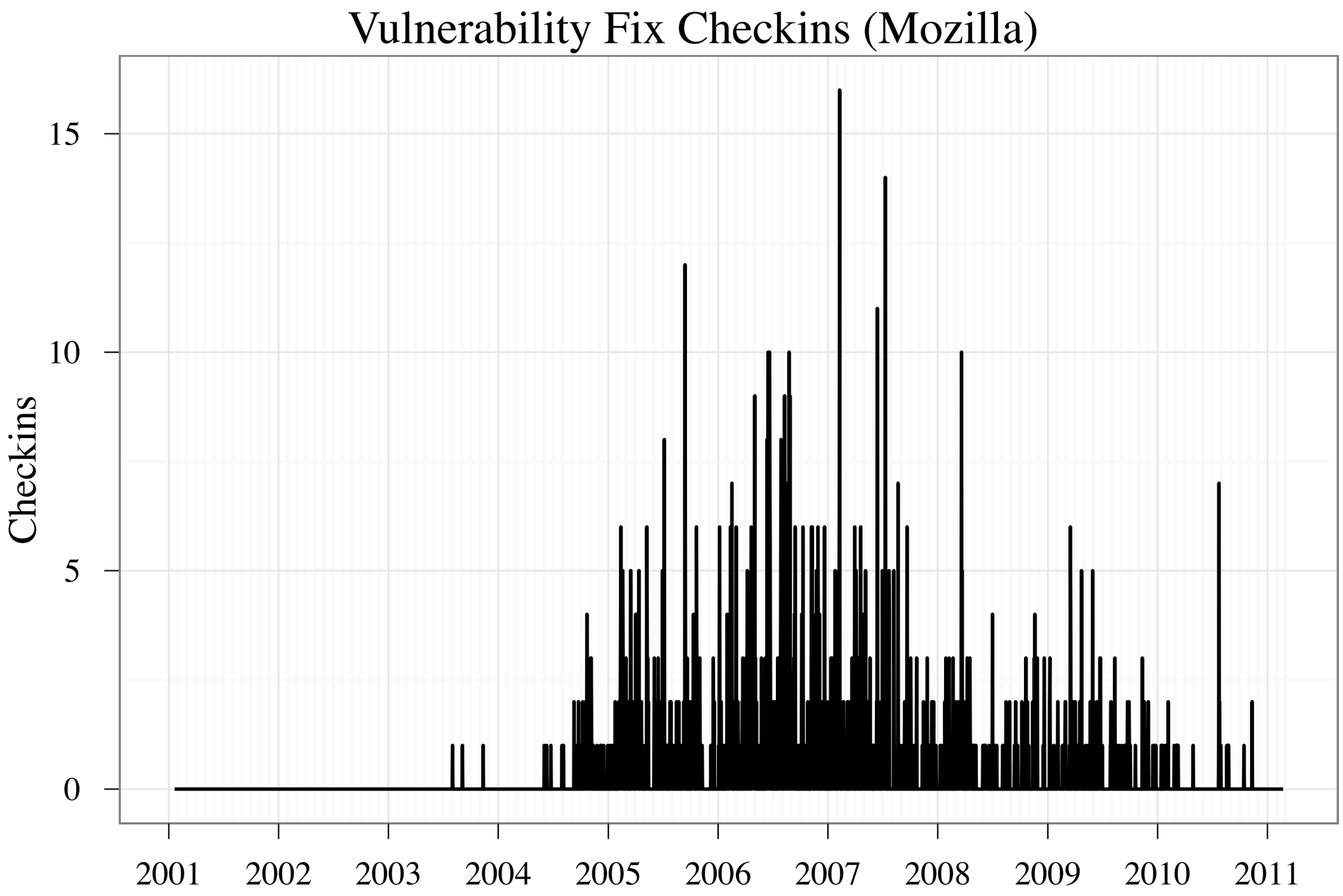
- Security information published through CVE
- No helpful links to bug reports
- Manual mapping of vulnerabilities to fixes
- Leaving out CVEs we can't assign
- Out of 100 CVEs, 66 remain

Image source: Apache foundation

Apache Tomcat

- From 2008 on, reports have revision number of fixing checkin :-)
- Before 2008, there is no link at all :-(
- Out of the 89 CVEs, only 21 remain
- Attribution absolutely certain

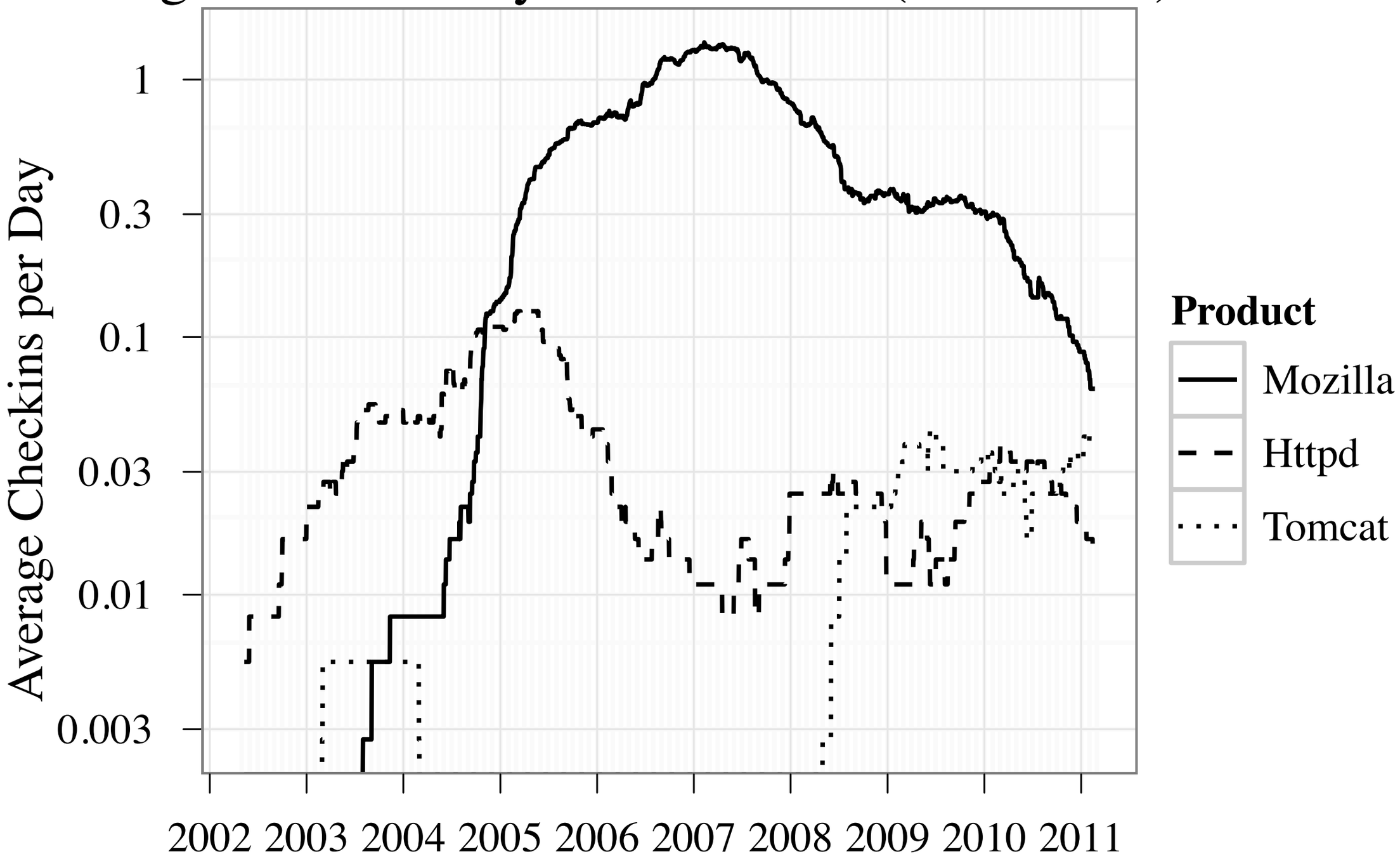
Image source: Apache foundation


Vulnerability Fix Checkins (Mozilla)
Checkins
15
10
5
0
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011

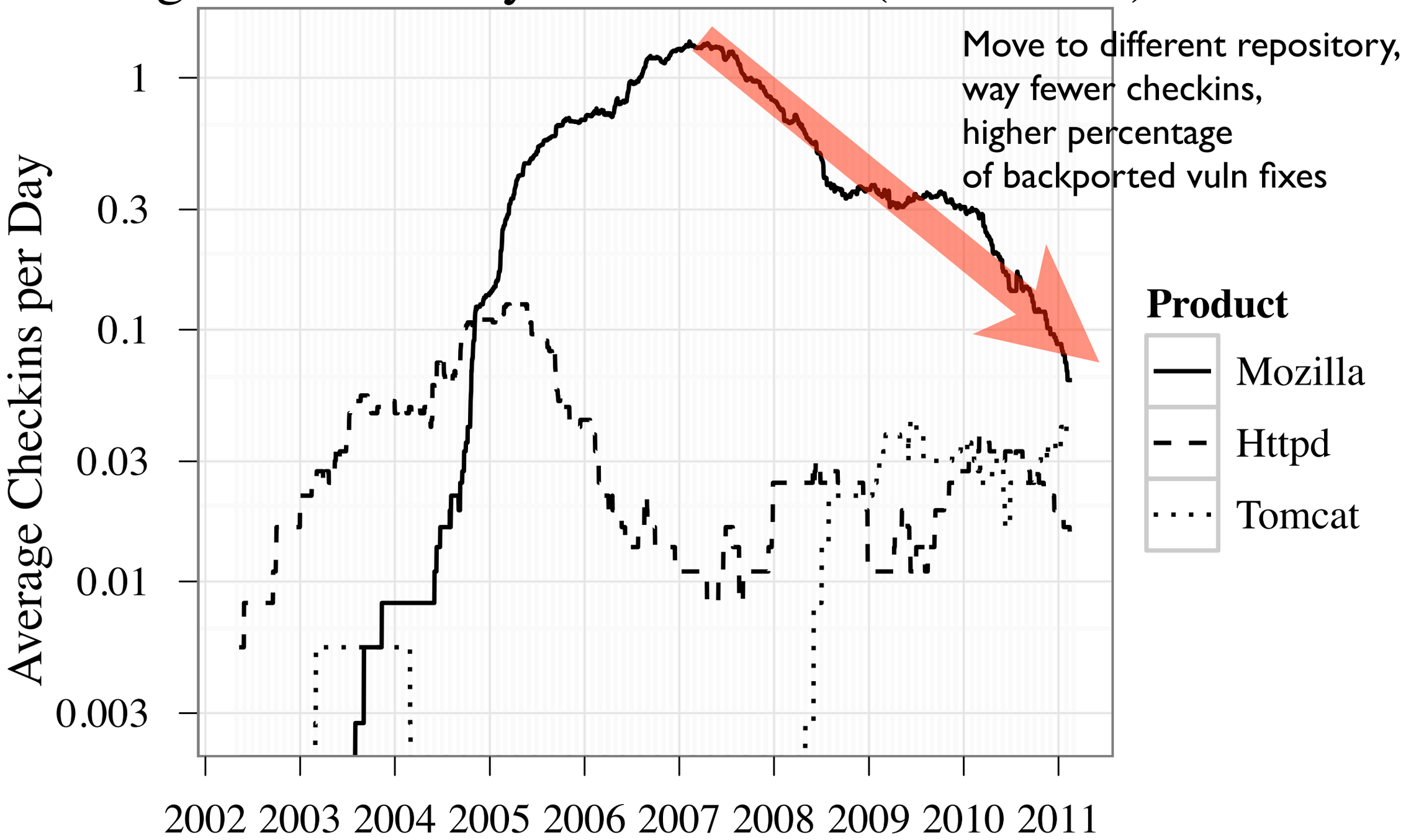
Moving Average

- Acts as a *low-pass filter*, removing high-frequency jiggling
- Smoothes out strong *day-to-day variations*
- Leaves overall *trend* (if any)
- Trend will appear with a *lag*
- We chose window size 365
  - Even in leap years

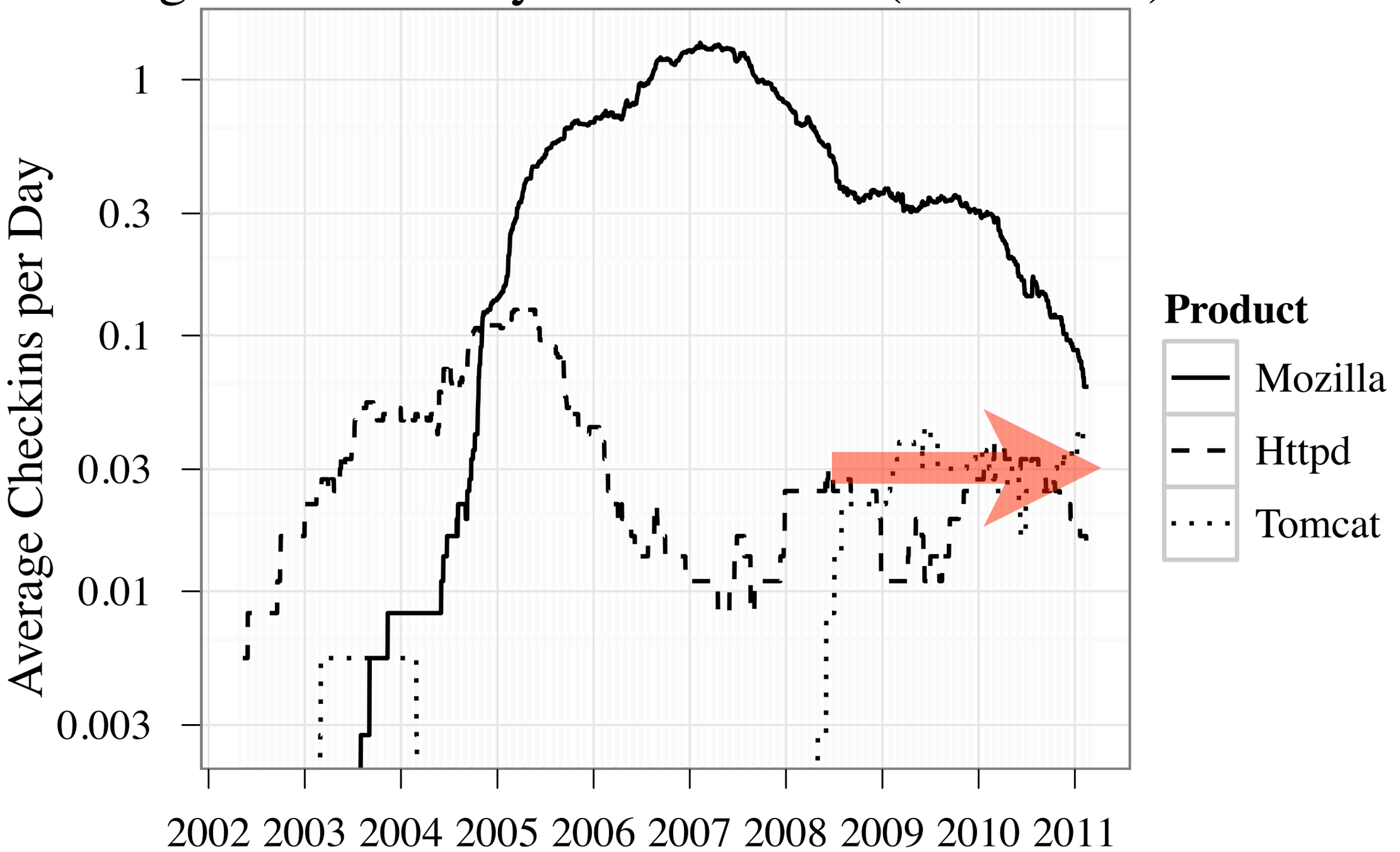
Average Vulnerability Fix Checkins (Combined)



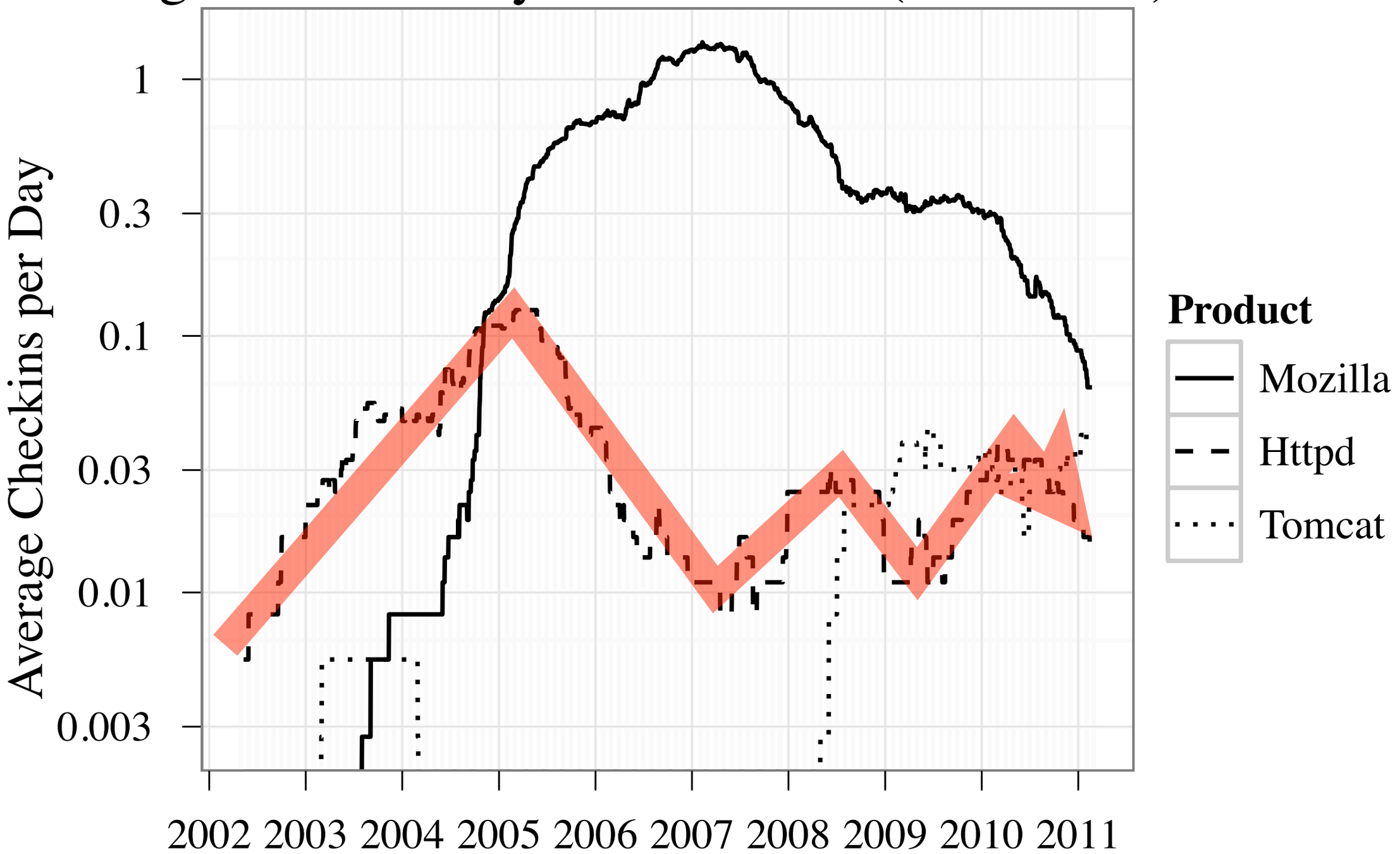
Average Vulnerability Fix Checkins (Combined)



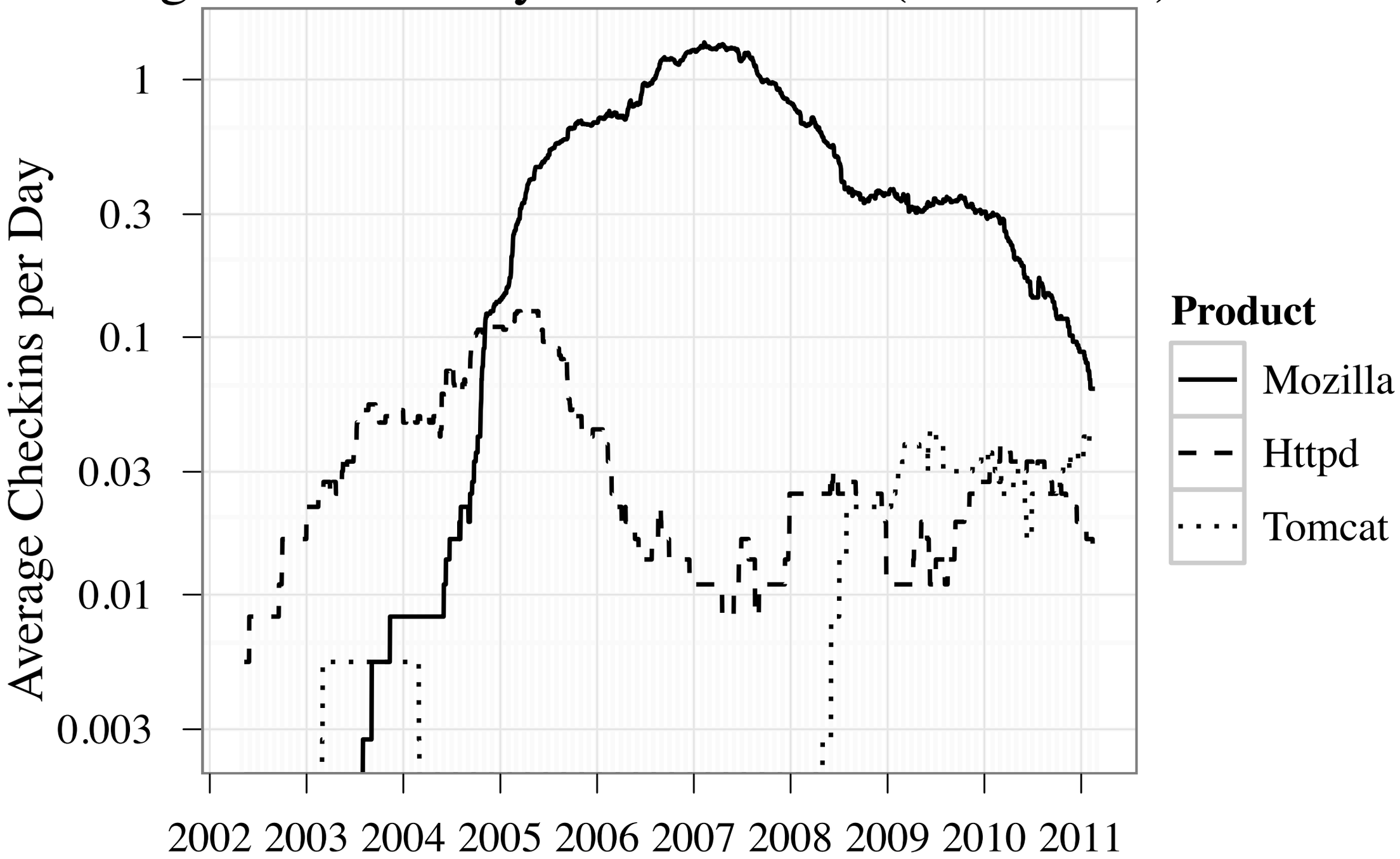
Average Vulnerability Fix Checkins (Combined)



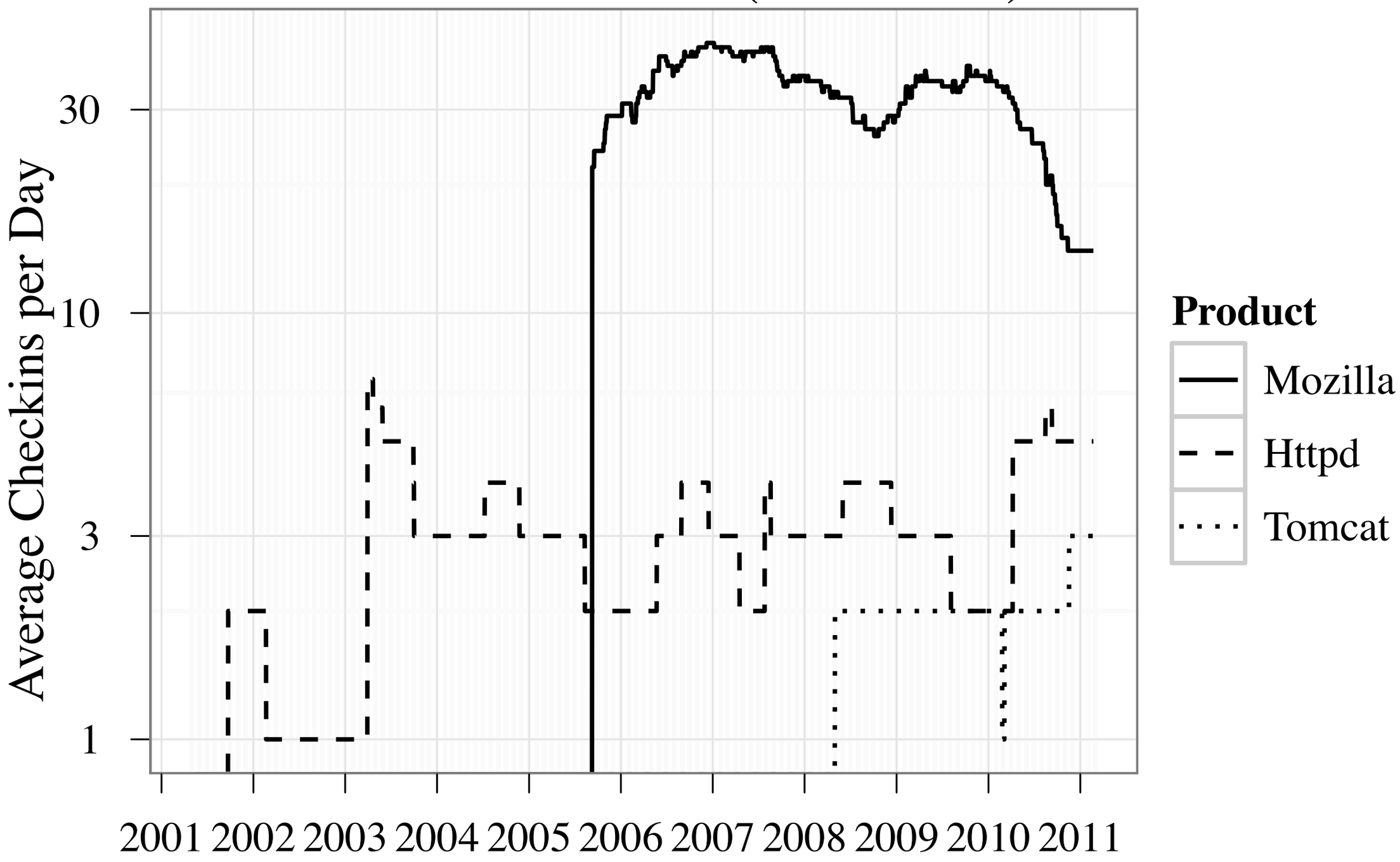
Average Vulnerability Fix Checkins (Combined)



Average Vulnerability Fix Checkins (Combined)



Number of Committers (Combined)



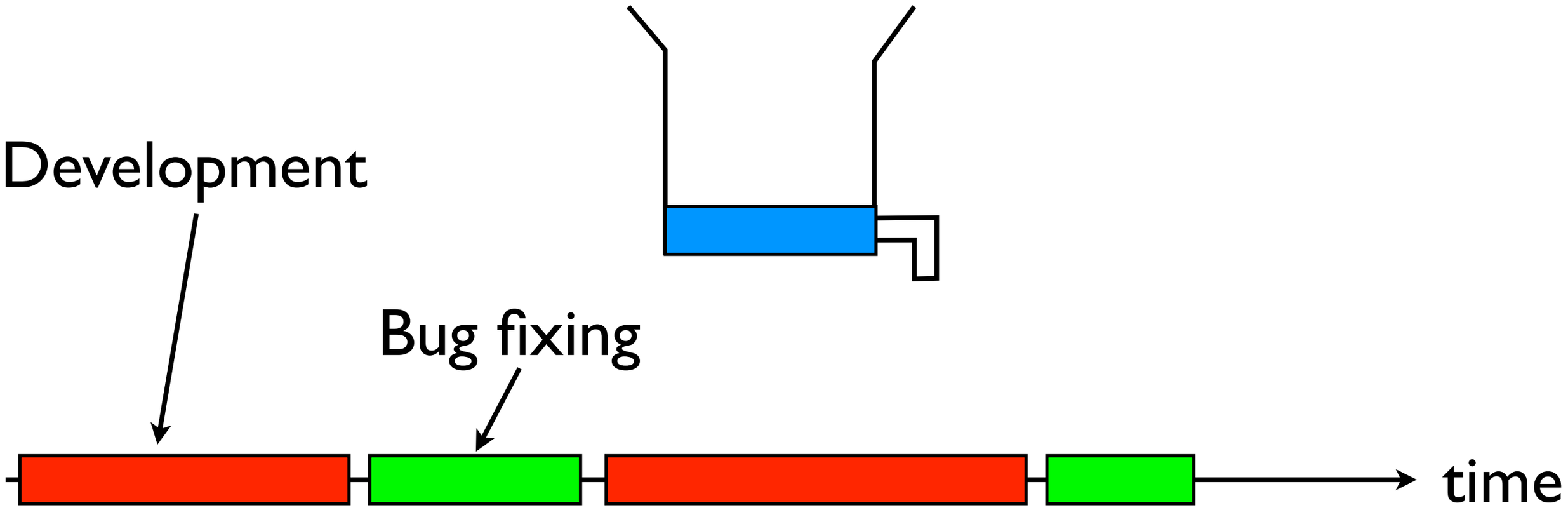
Why does the Standard Model not Fit?

- Standard model is for *static* situation
- Software development is *dynamic*, has *phases*
  - Next week is a feature freeze
  - In two months, writing that new feature
- Supply of easy vulnerabilities has not run out
- Why not? It's superficially plausible that they should!

Reservoirs

average length μ_{in}

average length μ_{out}



Reservoirs

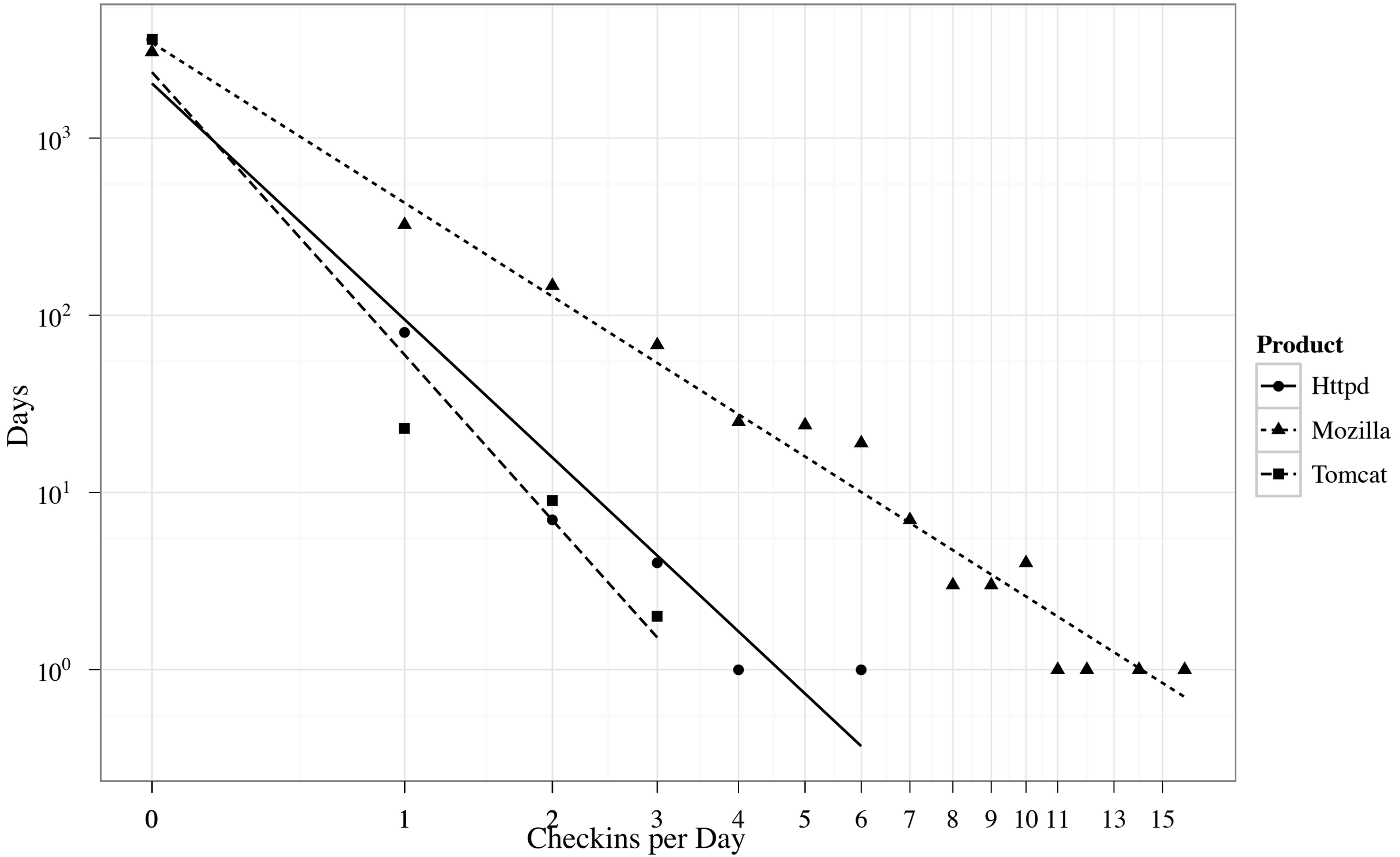
- There is a reservoir of vulnerabilities
- Phases of average length μ_{in} in which vulnerabilities are put into the reservoir (development)
- Phases of average length μ_{out} in which vulnerabilities are taken out of the reservoir (bug fixing)
- These phases alternate
- Not perfect, but more plausible than standard model!

Consequences

- "The number of [vulnerabilities] V will be heavy tailed with $P(V > x) \sim cx^{-(\alpha-1)}L(x)$, where c and α are constants and L a slowly varying function."
- Number of vulnerabilities unknown!

More Consequences

- Assume that in any given 365-day period, a constant fraction of the available vulnerabilities will be fixed, it follows that the number of days with a given number of checkins will also be heavy tailed
- #vulnerabilities fixed∝#vulnerabilities present
- Not a priori clear, but let's see what follows!


Days
10^3
10^2
10^1
10^0
0
1
2
3
4
5
6
7
8
9
10
11
13
15
Checkins per Day
Product
Httpd
Mozilla
Tomcat

Other Things in the Paper

- Is software security an arms race, where attackers and defenders have to work very hard just to maintain the status quo?
- Would lead to distribution between successive days with fixes obeying a power law
- No (or, if yes, lost in noise of other activity)

Neil Johnson, Spencer Carran, Joel Botner, Kyle Fontaine, Nathan Laxague, Philip Nuetzel, Jessica Turnley, and Brian Tivnan. *Pattern in escalations in insurgent and terrorist activity*. Science, **333**(6038):81–84, July 2011.

You Should not Believe Me!

- Make your own experiments on my data!
- Find bugs in my scripts!
- ftp://ftp.tik.ee.ethz.ch/pub/publications/WEIS2012/fixrates.tar.gz
- (Don't write this down, it's in the paper)

Personal Remarks

- Share your data and scripts!
- Many thanks to reviewer who pointed out problems with my “power law” analysis
- This work is not about surprises
- A scientific paper is not a news story

Conclusion

- Standard model not applicable for software development
  - Hence perhaps not so standard
- Reservoir model makes predictions that actually fit the data
  - Hence perhaps a better model
- Ultimately, phenomenon not understood