Zero Trust Theorem

We Are Developers World Congress 2019, Berlin

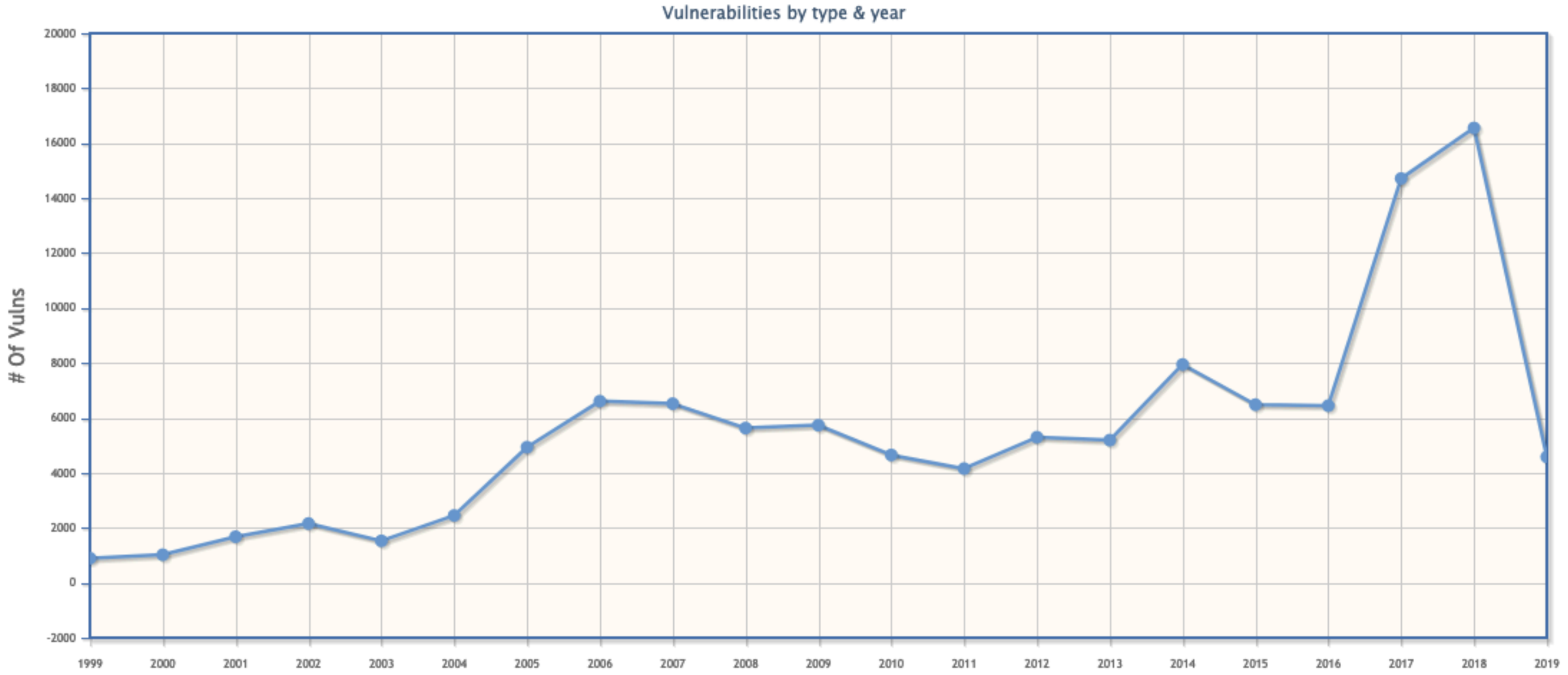
Andrzej Dyjak, Head of AppSec @ AFINE

- Currently: **Head of Application Security @ AFINE (**https://afine.pl/**)**
- Previously: Security Architect, Security Researcher, and Software Engineer
- In the past I also found critical low-level vulnerabilities in software from major software vendors


Preludium

- What will we cover? (hint: AppSec)
- How will we cover it? (hint: real-world case studies)


I
IF IT
AIN'T
BROK
DON'T
FIX IT


Vulnerabilities by type & year
Of Vulns
20000
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
-2000
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019

Web Applications

Case Studies 

- Choices are endless… HackerOne, BugCrowd, etc
- Developers usually understand vulnerabilities on app-level
- Well described in documents such as OWASP Top 10 or Application Security Verification Standard

Mitigations 

- Use frameworks because “*Given a thousand eyes, all bugs are shallow.*” — Linus (e.g. Ruby on Rails)
- Follow standard ways of *doing things* (e.g. The Rails Way for RoR)
- Secure SDLC
  - Test according to well-known standards (e.g. OWASP ASVS)
  - Include Security into DevOps (making it DevSecOps)
  - Adoption of a Secure SDLC framework (MS SDL, OWASP SAMM, Synopsys’ BSIMM)


WE NEED TO GO
DEEPER

External Components

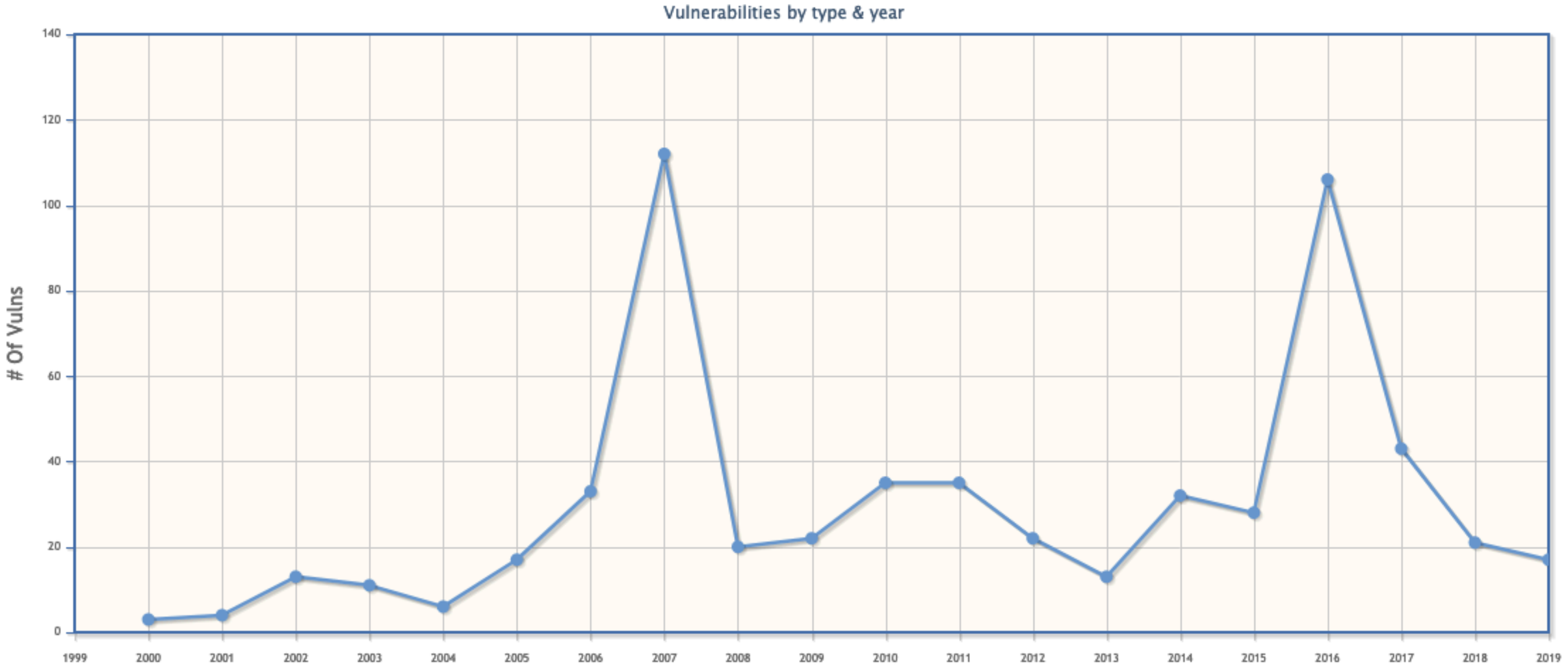
Case Studies 

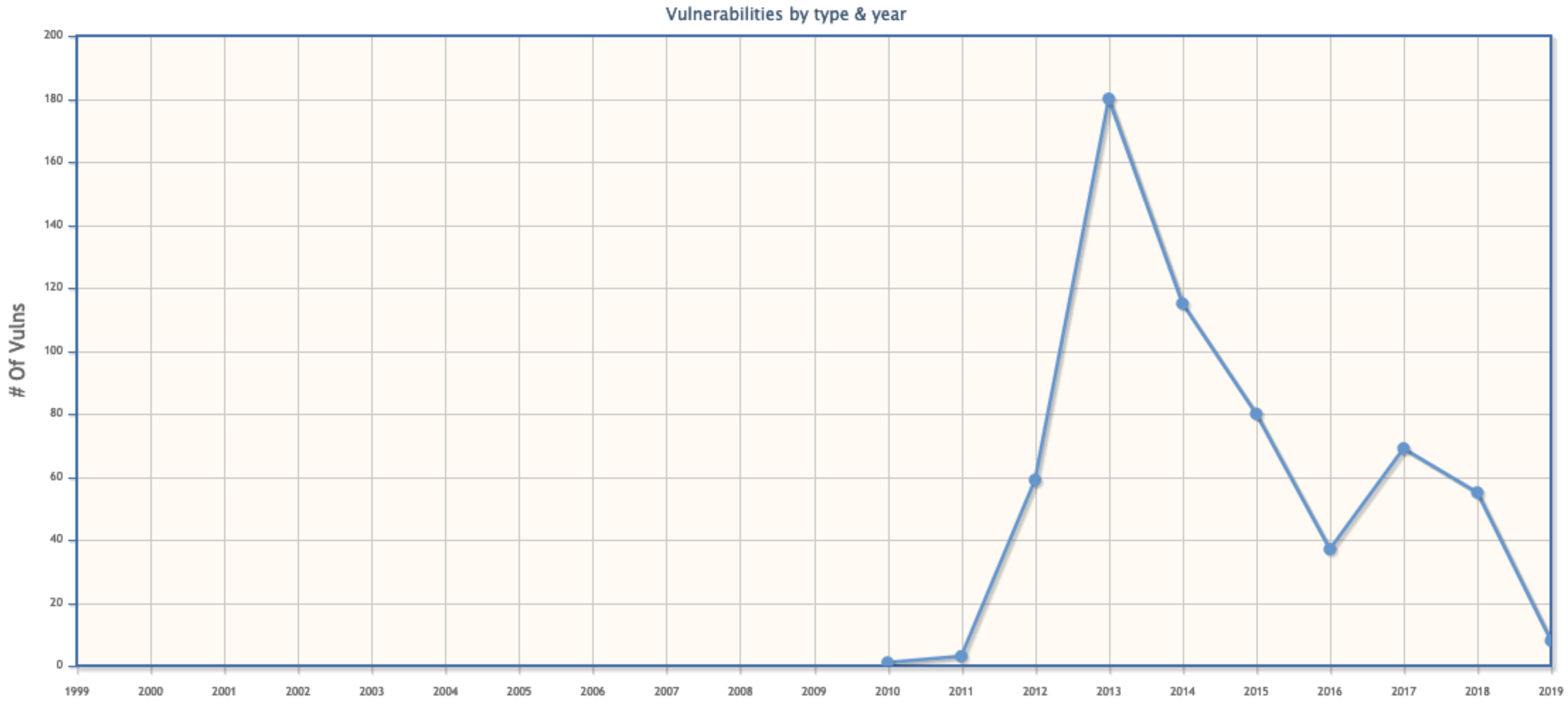
- OS command injection in the way GraphicsMagick utility was used — Imgur
- Memory disclosure bugs in ImageMagick — vulnerable versions identified on servers from Dropbox and Yahoo! (among others)

Mitigations 

- An informed choice of external components
  - Small attack surface = small risk
- Apply the *Principle of Least Privilege* (e.g. external binaries should be inside of a sandbox)

Interpreters
&
Virtual Machines (JVM, CLR, etc)


Vulnerabilities by type & year
Of Vulns
140
120
100
80
60
40
20
0
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019


Vulnerabilities by type & year
Of Vulns
200
180
160
140
120
100
80
60
40
20
0
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019

Case Studies 

- Deserialization of a `cookie` parameter, and memory corruption within `unserialize()` in native PHP (Zend) — PornHub
- “The worst bug bounty ever” — an expensive romance between Shopify & mruby
- “Exposing Hidden Exploitable Behaviors in Programming Languages Using Differential Fuzzing” — *interesting* behaviours in interpreters
- My own vulnerability research for major interpreters (for fun & no profit)

	PHP (php-7.1.1-asan)	HHVM (hhvm-3.15.6-dev)	Ruby (mri-2.4.0-asan)	Python (cpython-2.7.13-asan)
EXPLOITABLE	**58**	**35**	**74**	**2**
PROBABLY_EXPLOITABLE	**8**	**0**	**0**	**2**
PROBABLY_NOT_EXPLOITABLE	**8**	**0**	**2**	**4**
UNKNOWN	**12**	**5**	**5**	**3**

Mitigations 

- Apply *Principle of Least Privilege*
- Problematic functions should be banned
  - **Softcore:** Code Review / SCM level 😀
  - **Hardcore:** Interpreter level (delete specific functions from the source code of the underlying interpreter then recompile 😎)

Recompile? 🤔

Compilers

Case Studies 🔬

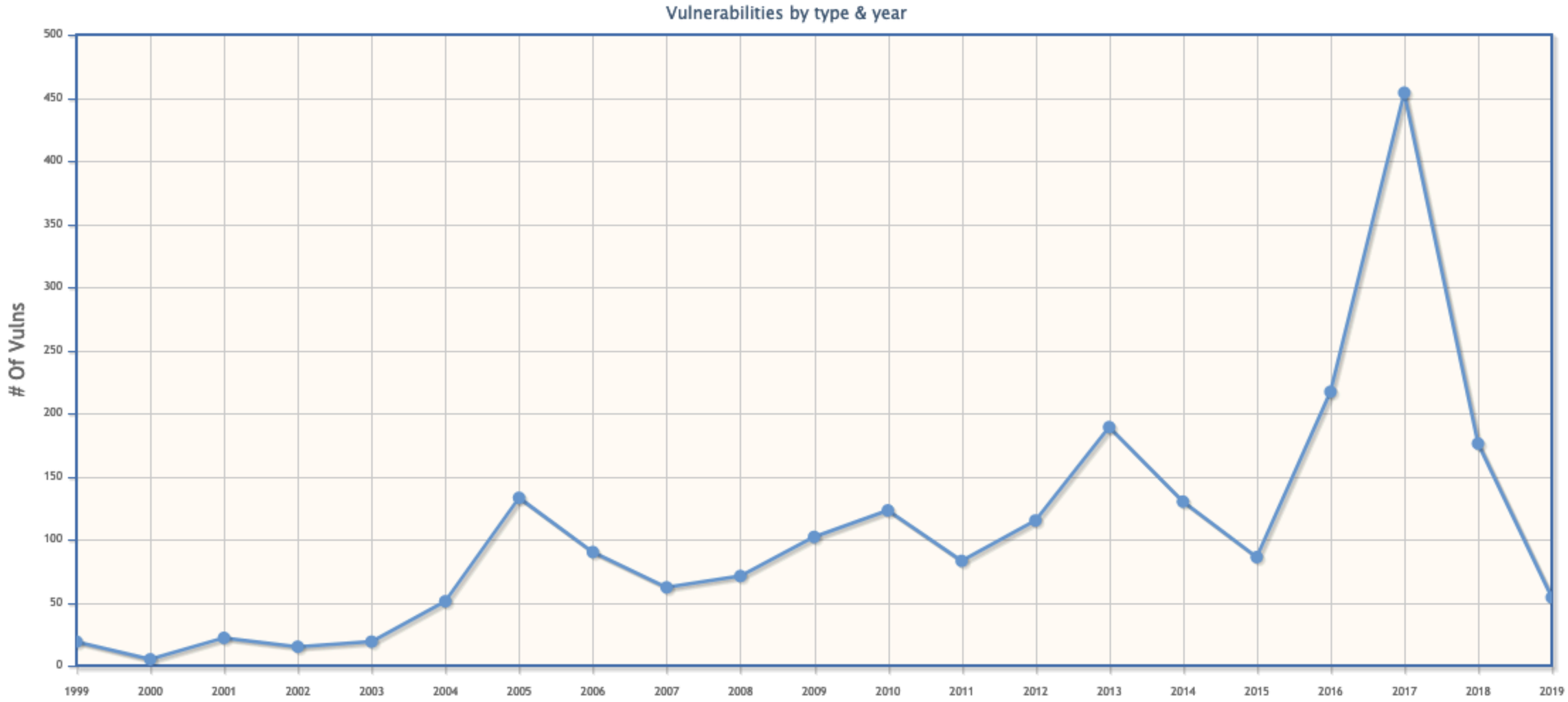
- “Reflections on Trusting Trust” — Ken Thompson
- CVE-2018-1037 — .PDB Heap Memory Disclosure in Visual Studio by j00ru (Google Project Zero) 🤘

Mitigations 

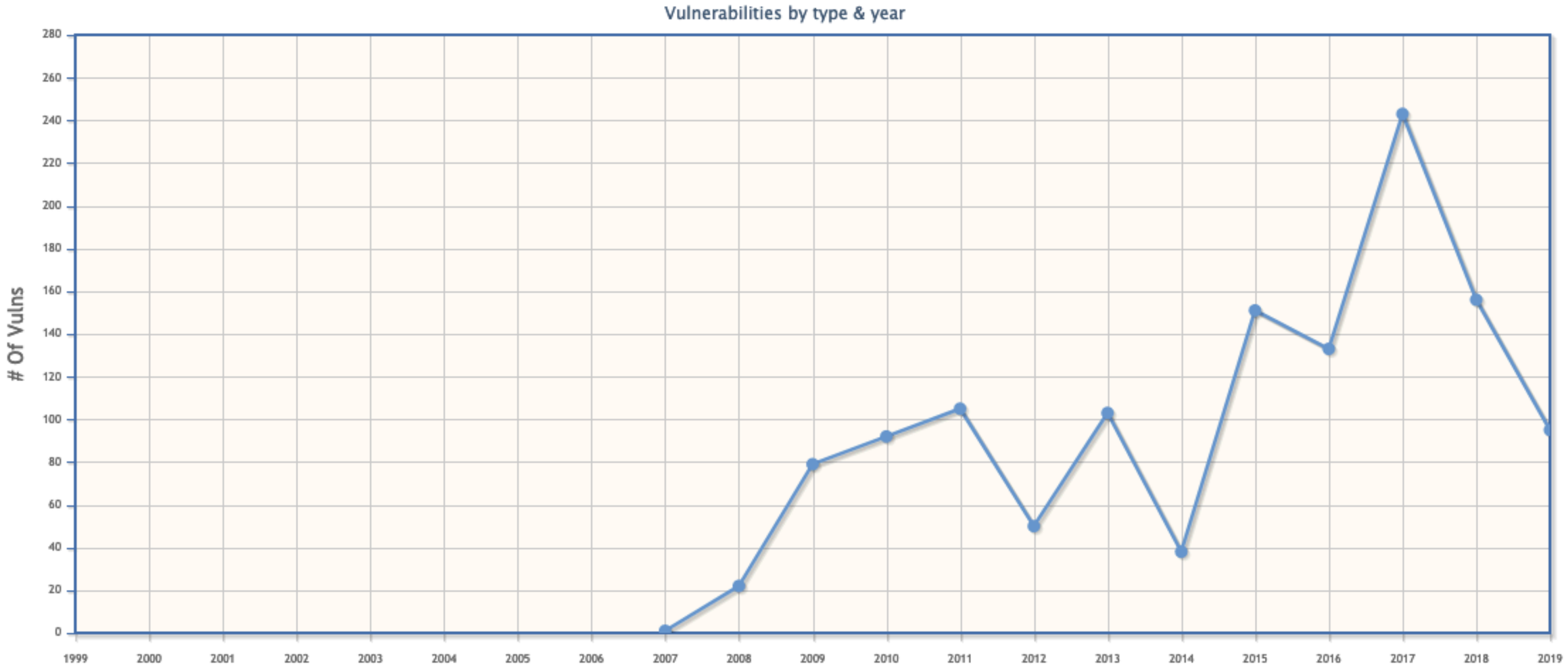
- Lack of active defense
  - Integrity monitoring (passive) of our environment

Operating Systems

Linux


Vulnerabilities by type & year
Of Vulns
500
450
400
350
300
250
200
150
100
50
0
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019

Windows*



*** Windows Server 2008**

Case Studies 🔬

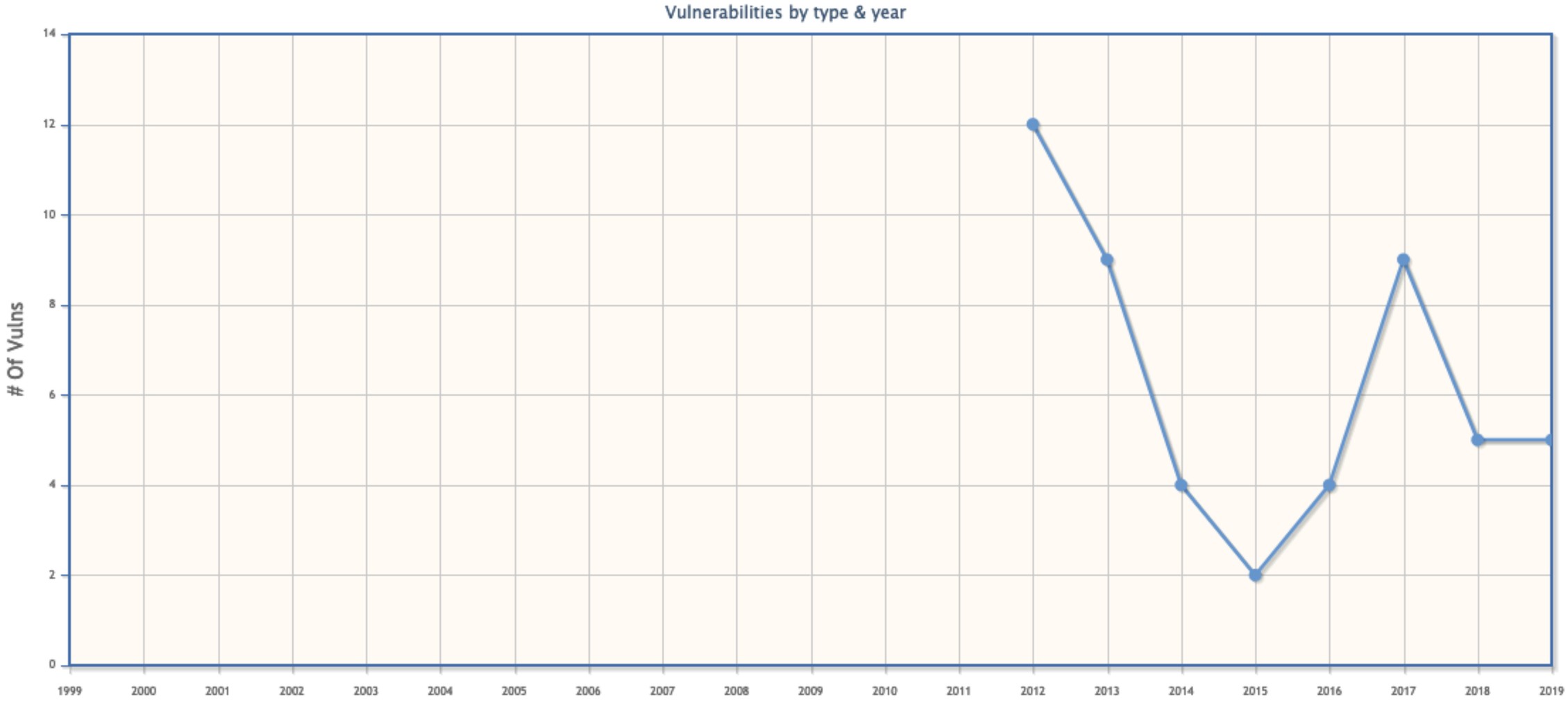
- CVE-2016-5195 — DirtyCOW
- CVE-2010-0232 — KiTrap0D from Tavis Ormandy (Google)
- CVE-2012-0217 (and its younger brother CVE-2006-0744) — Intel SYSRET found in 2012 by Rafał Wojtczuk (InvisibleThingsLab)
- CVE-2018-8897 — POPSS/MOVSS

Mitigations 

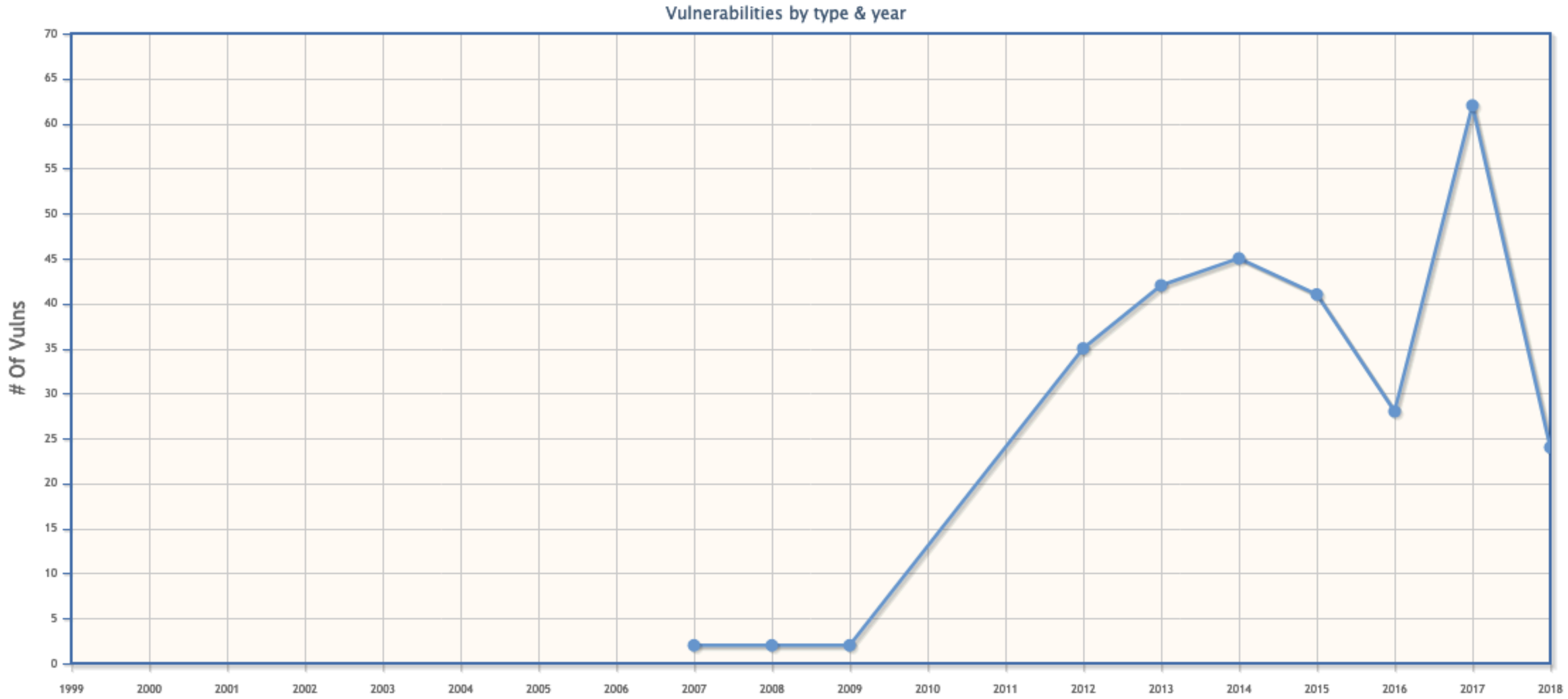
- Patch management policy
- Hardening
  - Best practices (e.g. NIST, CIS Benchmarks)
  - Additional defense mechanisms (e.g. Linux - grsecurity, LKRG; Windows - EMET for old Windows or WDEG for newer Windows releases)

Hypervisors

VMware (ESXi)


Vulnerabilities by type & year
Of Vulns
0
2
4
6
8
10
12
14
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019

XEN


Vulnerabilities by type & year
Of Vulns
70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018

Case Studies 🔬

- Cloudburst — *guest escape* in VMware from 2009 (via SVGA)
- Pwn2Own — Olympic Games in software hacking 🤓
  - 2016 — Virtualisation added to the competition
  - 2017 — 2 teams successfully escaped VMware
  - 2019 — VirtualBox escape x 2, VMware escape x 1

Mitigations 🛡️


THIS IS FINE.

Hardware

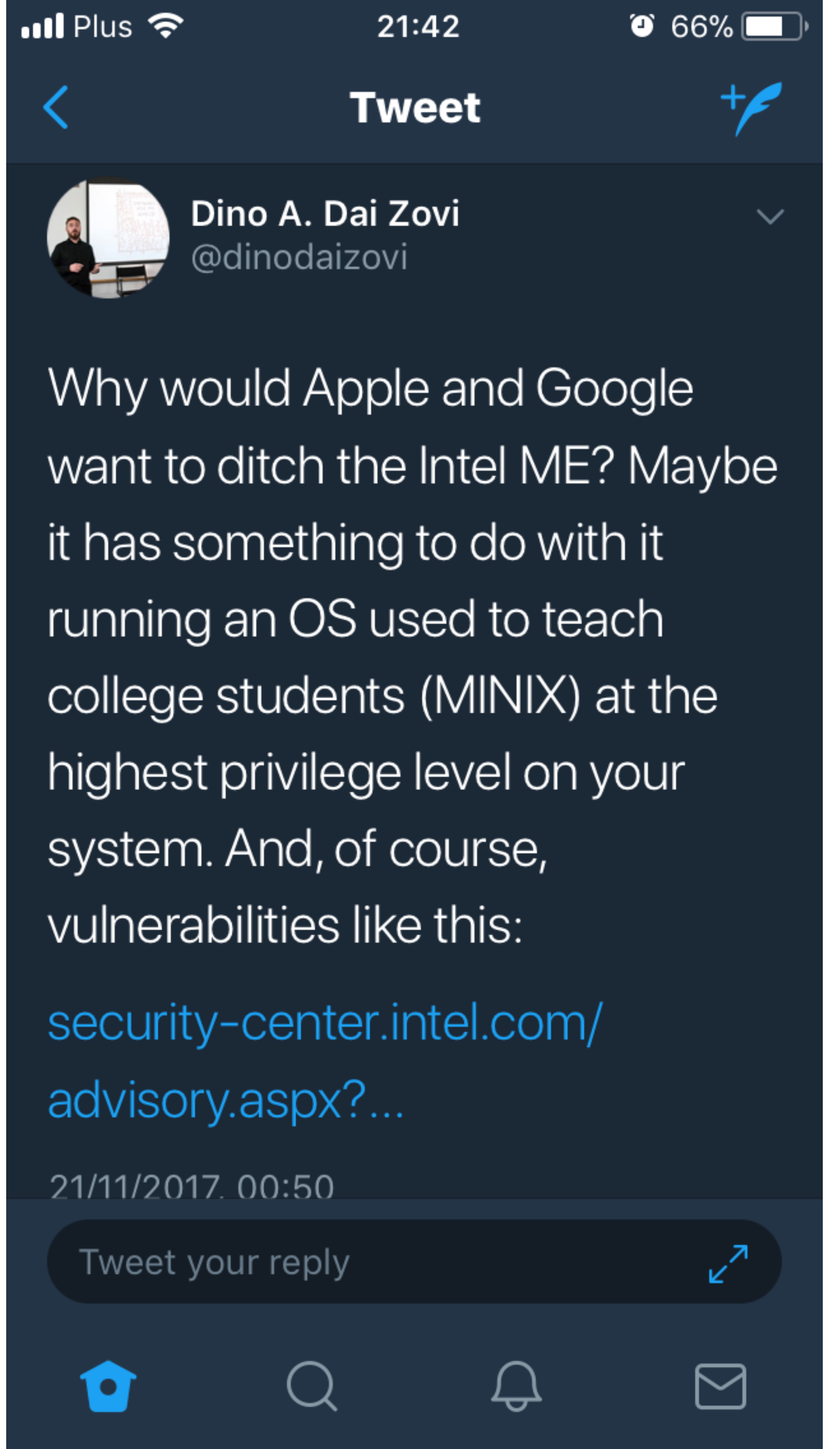
Case Studies 🔬 — CPU 1/2

- Bugs
  - Pentium FDIV bug — Intel - $$$ = 😢
  - AMD microcode security update — Robert Święcki during kernel fuzzing on a desktop
  - Meltdown & Spectre — Jann Horn (Project Zero) et al

Case Studies 🔬 — CPU 2/2

- Features? 🤔
  - sandsifter — CPU fuzzing on BlackHat 2017 by Christopher Domas
  - Intel-SA-00086 — bugs in Intel Management Engine (ME)

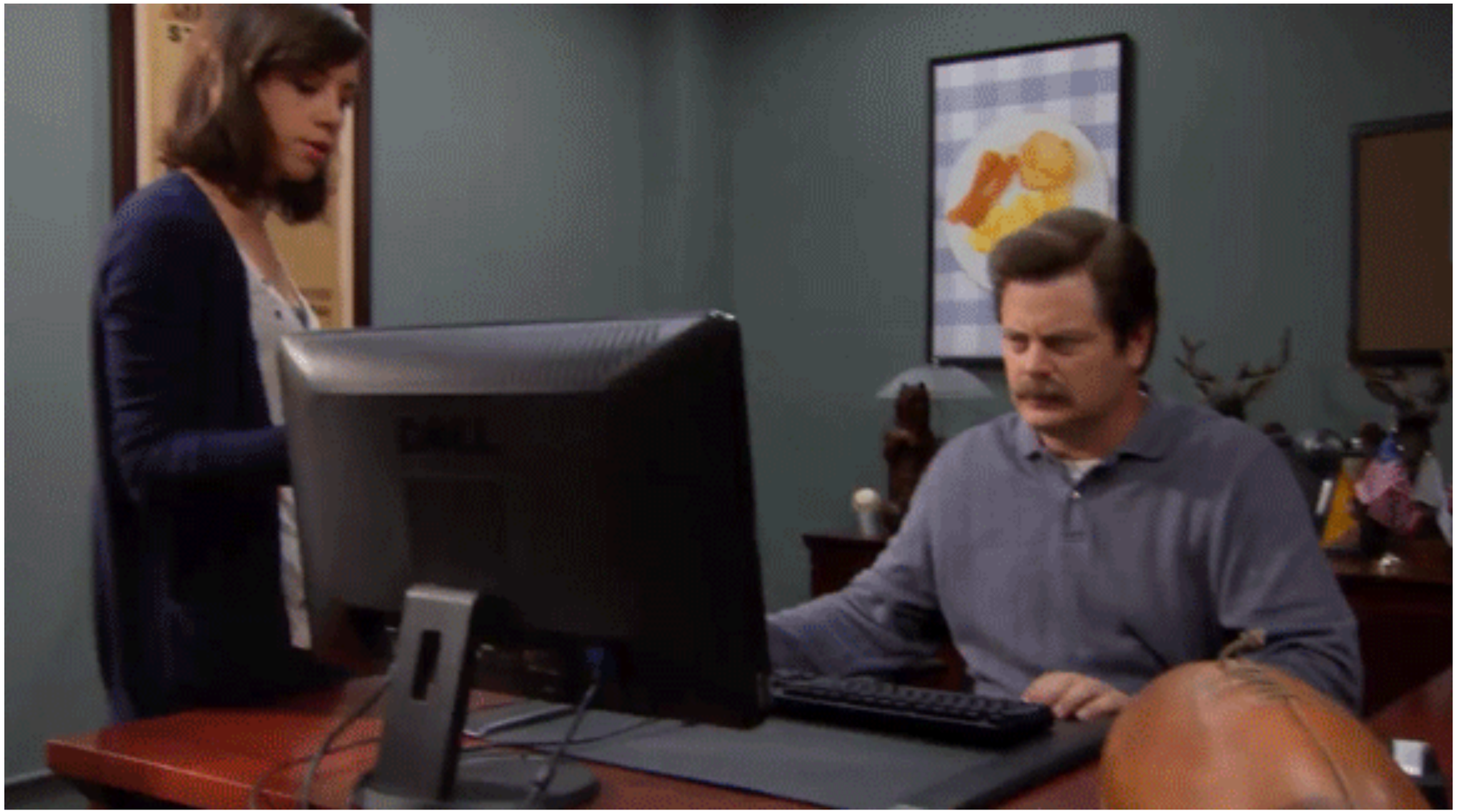

Plus
21:39
67%
Tweet
Nikolaj Schlej
@NikolajSchlej
Just a humble reminder to everyone freaking out by INTEL-SA-00086 aka total ME takeover: ME is a core of fTPM 2.0, BootGuard, SGX, PAVP, ICC, DAL and who knows what else on modern Intel platforms, so you you want to freak out - do it harder! ;)
21/11/2017, 00:32
101 Retweets 143 Likes
Tweet your reply


Plus
21:42
66%
Tweet
Dino A. Dai Zovi
@dinodaizovi
Why would Apple and Google want to ditch the Intel ME? Maybe it has something to do with it running an OS used to teach college students (MINIX) at the highest privilege level on your system. And, of course, vulnerabilities like this:
security-center.intel.com/advisory.aspx?...
21/11/2017, 00:50
Tweet your reply

Case Studies 🔬 — RAM

- RowHammer — Original idea by CMU and Intel researchers, then pushed forward by Thomas Dullien et al (Project Zero). Further research done by academia. Timeline:
  - First (2015) — desktops (local)
  - Later (2016) — mobiles (local) and VM-to-VM attacks (“local”)
  - Recently (2018) — mobiles (remote!) and servers in the cloud (remote!)

Mitigations



Summary 

- Software is broken all the way down
- Hardware is broken and it’s only the beginning
- Good practices on each layer lower the risk but will never eliminate it
- **Security is a process, not a product**

👋

andrzej@dyjak.me

@andrzejdyjak

https://afine.pl