TRAIL OF BITS Defense Guided by Experience

From One Ivory Tower to Another: Wish Listing for Filling the Gaps in Information (In)Security

Vincenzo Iozzo

Director of Security Engineering
Trail of Bits, Inc

My ivory tower

The security community is fixated on persistence

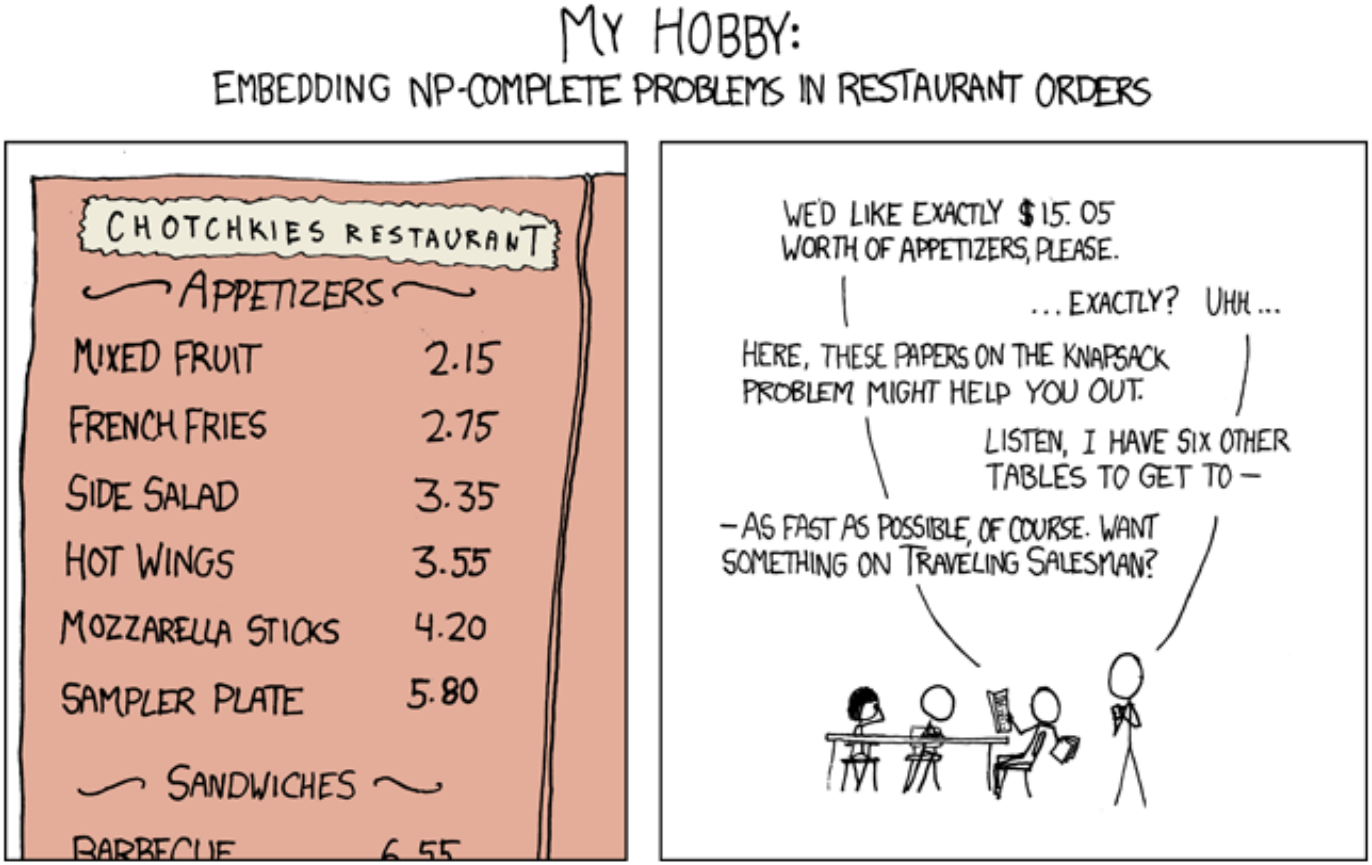
The mantra: “whoever scores is right”

Technical elegance is highly valued

Essentially: results trump (almost) everything

This happens a lot..

..while reading academic papers



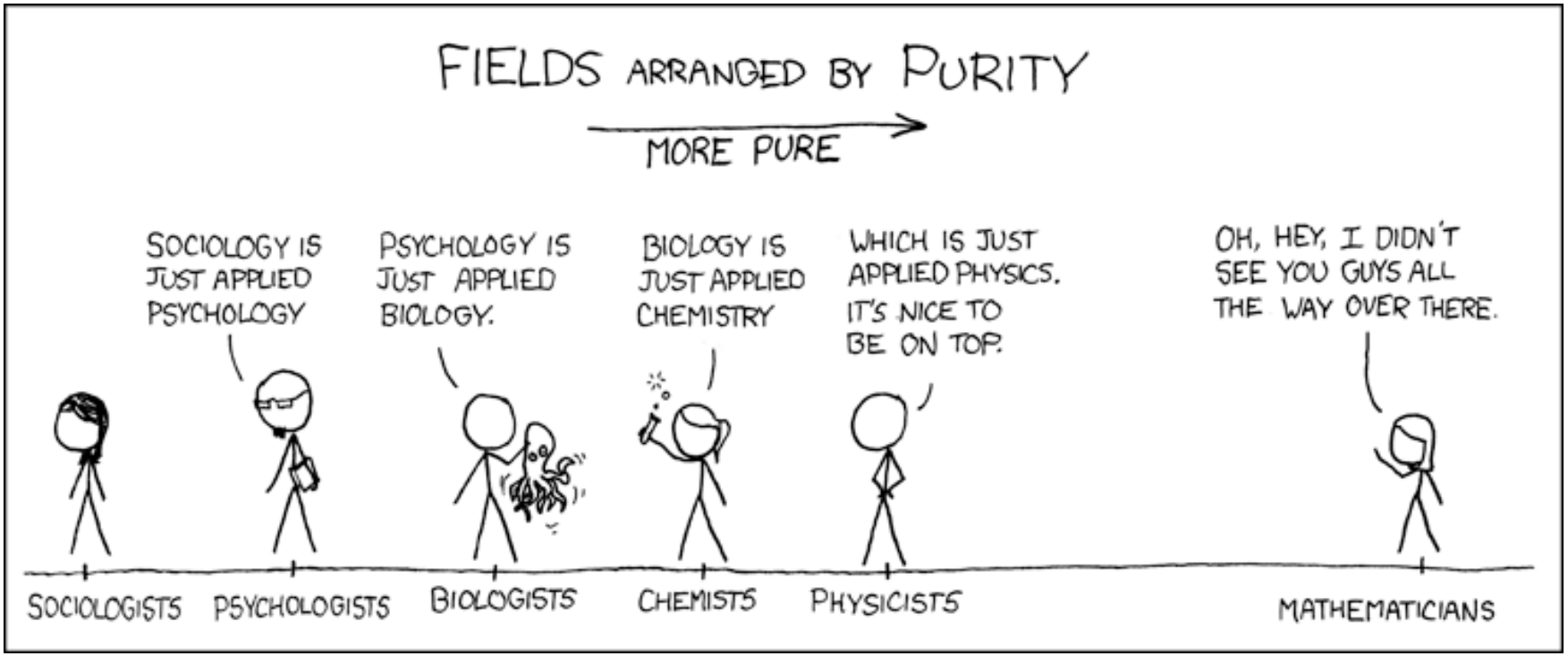
The NSA and your ivory tower

(U) Three of the last four sessions were of no value whatever, and indeed there was almost nothing at Eurocrypt to interest us (this is good news!). The scholarship was actually extremely good; it's just that the directions which external cryptologic researchers have taken are remarkably far from our own lines of interest.

The NSA says..

(U) I think I have hammered home my point often enough that I shall regard it as proved (by emphatic enunciation): the tendency at IACR meetings is for academic scientists (mathematicians, computer scientists, engineers, and philosophers masquerading as theoretical computer scientists) to present commendable research papers (in their own areas) which might affect cryptology at some future time or (more likely) in some other world. Naturally this is not anathema to us.

Is this it?


FIELDS ARRANGED BY PURITY
MORE PURE
SOCIOLOGY IS JUST APPLIED PSYCHOLOGY
PSYCHOLOGY IS JUST APPLIED BIOLOGY.
BIOLOGY IS JUST APPLIED CHEMISTRY
WHICH IS JUST APPLIED PHYSICS. IT'S NICE TO BE ON TOP.
OH, HEY, I DIDN'T SEE YOU GUYS ALL THE WAY OVER THERE.
SOCIOLOGISTS
PSYCHOLOGISTS
BIOLOGISTS
CHEMISTS
PHYSICISTS
MATHEMATICIANS

How do we make the relationship better?

TRAIL OF BITS

Note

Given the audience we will mostly focus on FUD (or “the future” if you prefer)

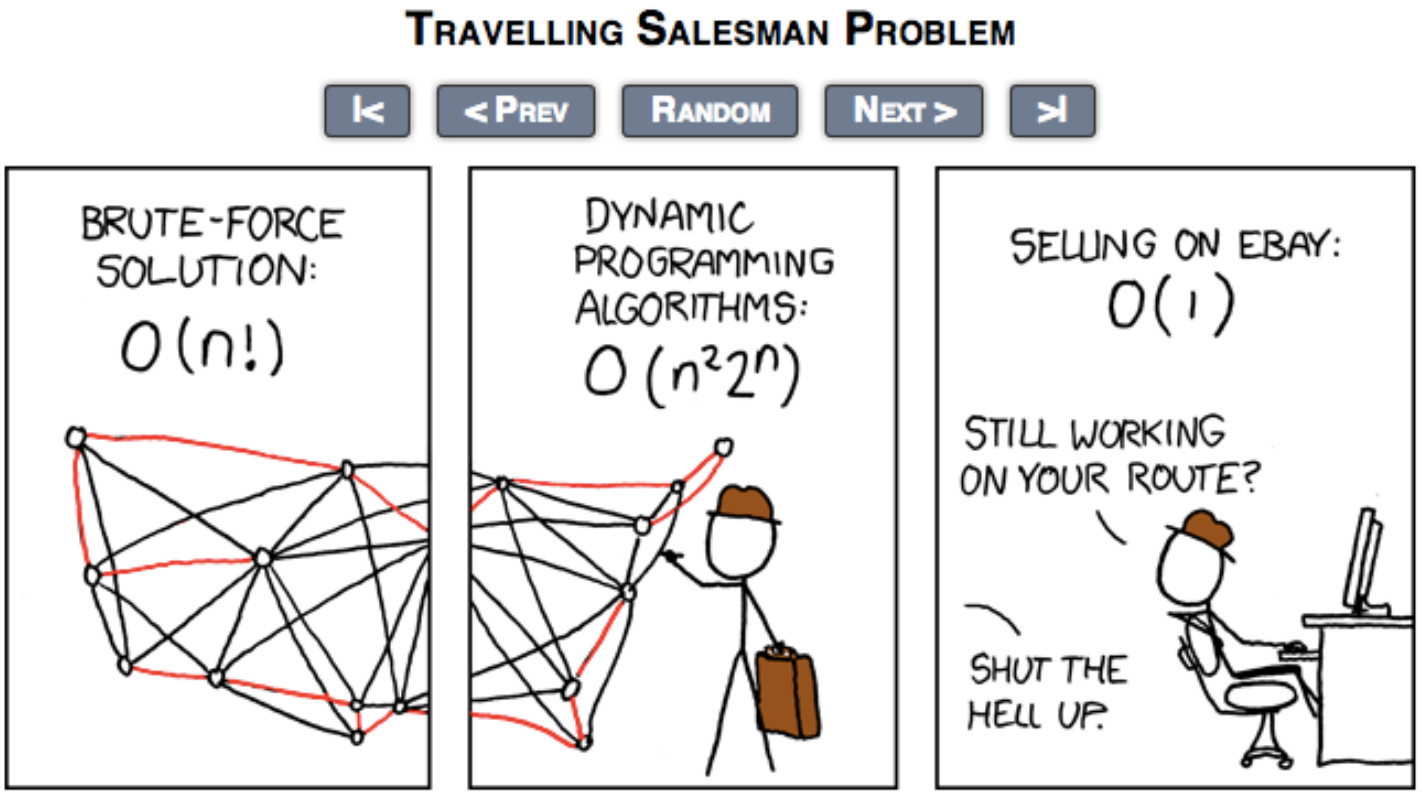
TRAIL OF BITS

Problem

Most of the problems we deal with are either intractable or undecidable in the general case

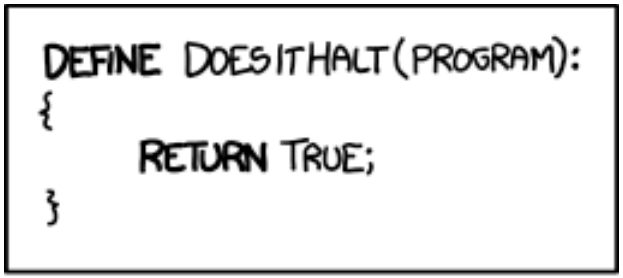
Small, overlooked fact

Exploitation is successful due to specificity



Thesis/Solution

Get specific, get practical



THE BIG PICTURE SOLUTION
TO THE HALTING PROBLEM

Mostly two topics

Vulnerability discovery

Exploitation

Vulnerability Discovery

Success stories

HAVOC/HAVOC-LITE (Julien Vanegue et al)

Bochspwn (Jurczyk et al)

Chucky (Fabian Yamaguchi et al)

TRAIL OF BITS

HAVOC-LITE

Check reference counting issues in COM interfaces

Solution: Add a ghost property to the model of the object to check for QueryInterface correctness

Bochspwn

Ring0/Ring3 Race conditions in Windows

Solution: Instrument a Windows machine with Bochs to log memory access. Enhance the analysis by:

1) only analyzing reads > 1 byte
2) same-size consecutive reads
3) remove known useless patterns
4) more stuff..

Chucky

Find missing checks

Solution: Automate the natural "pattern matching" work that bug-hunters do through anomaly detection based on other instances of similar code snippets inside the application

TRAIL OF BITS

So.. What to focus on?

Novel (this goes without saying, right?)

Hard

Driven by real world-data

Practical

TRAIL OF BITS

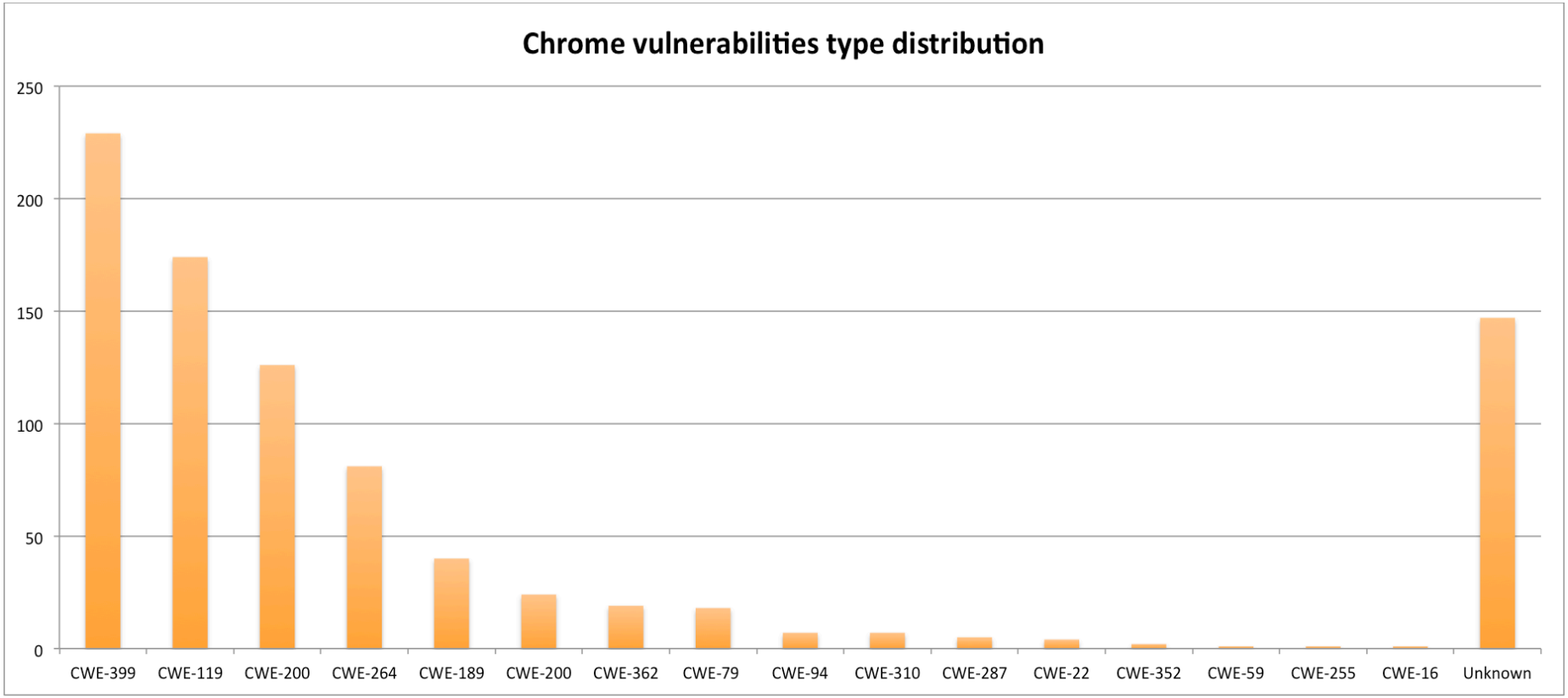
What's hard?

Bugs dependent on precise heap modeling

Concurrency

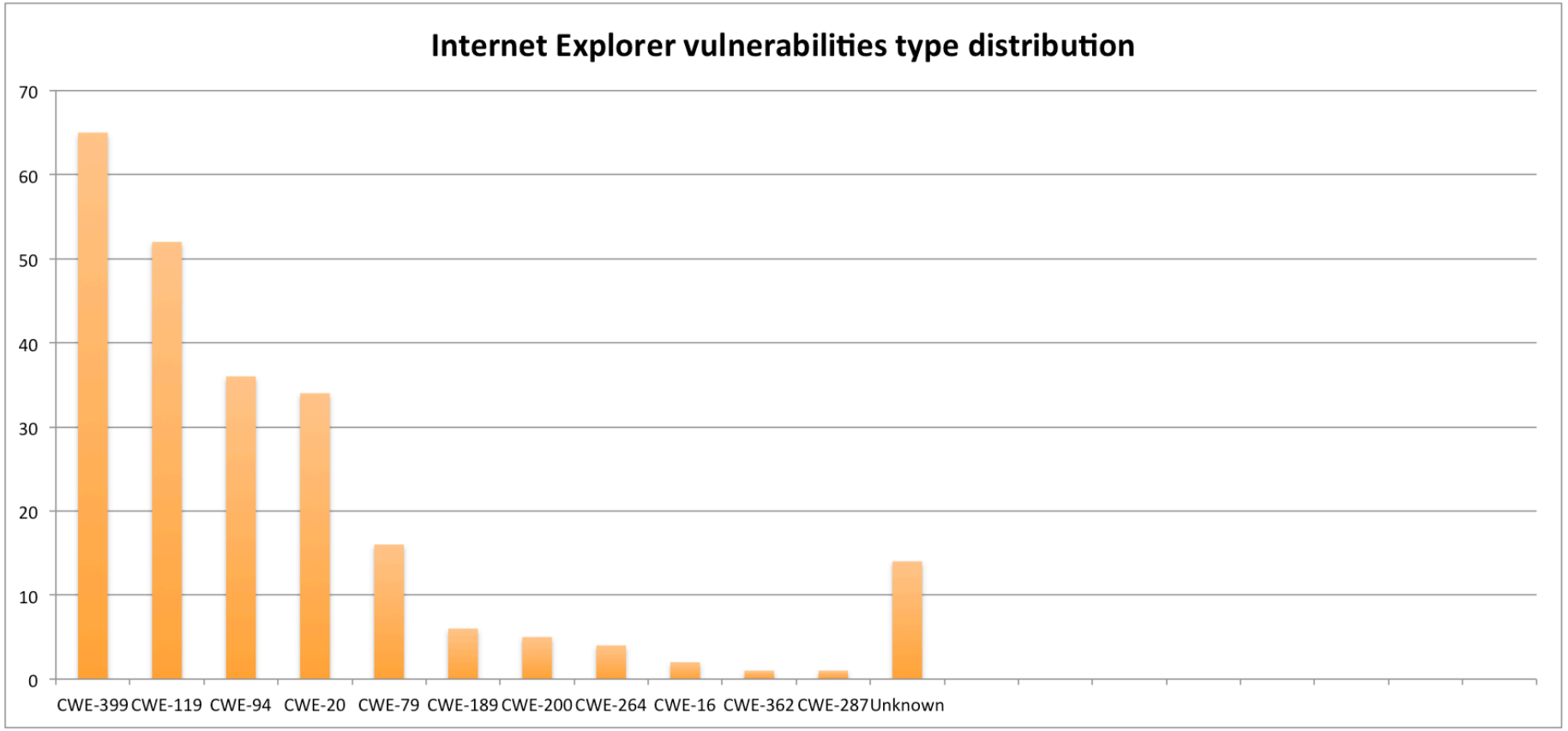
Logic bugs

Some data - Chrome

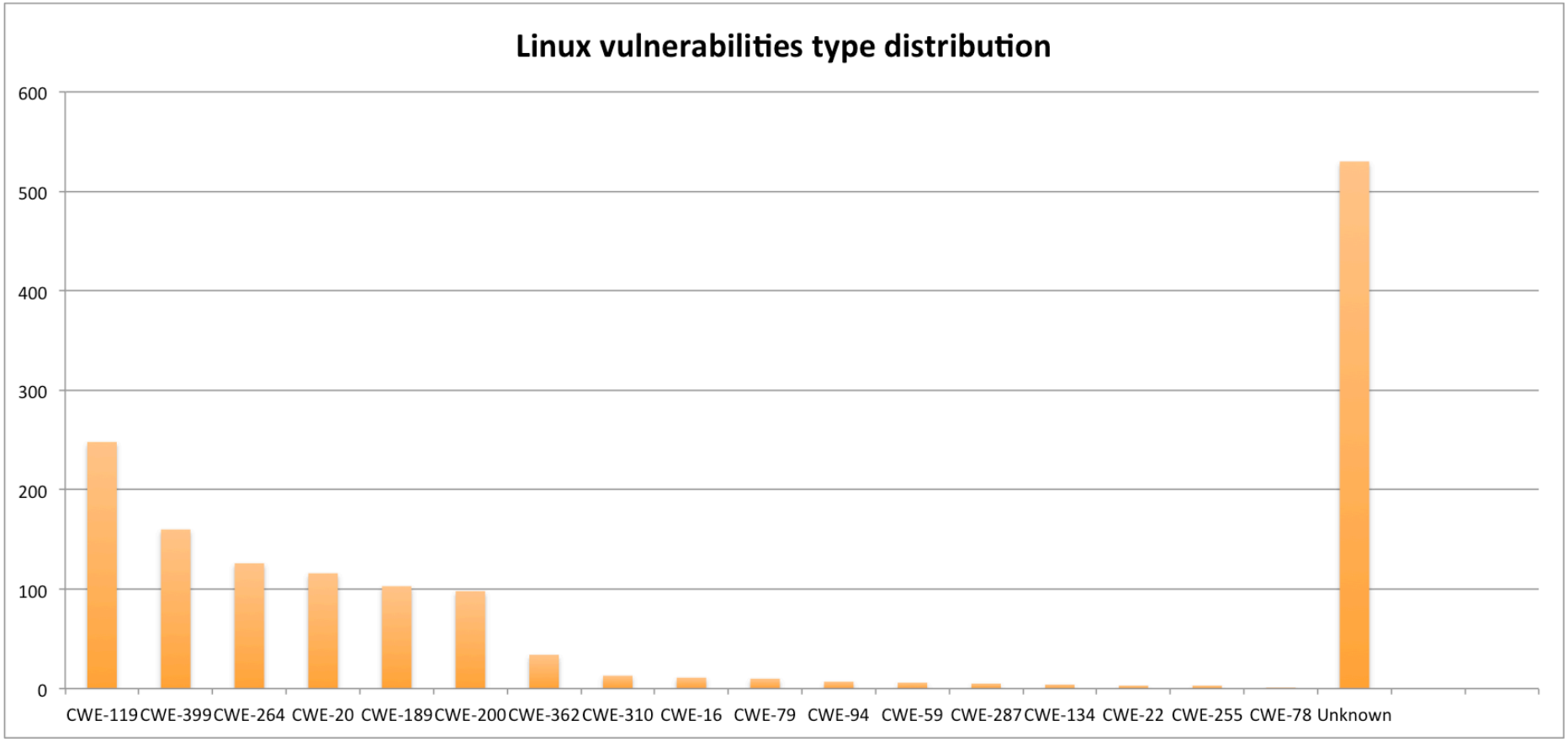


TRAIL OF BITS

IE



Linux kernel



Some data from MS

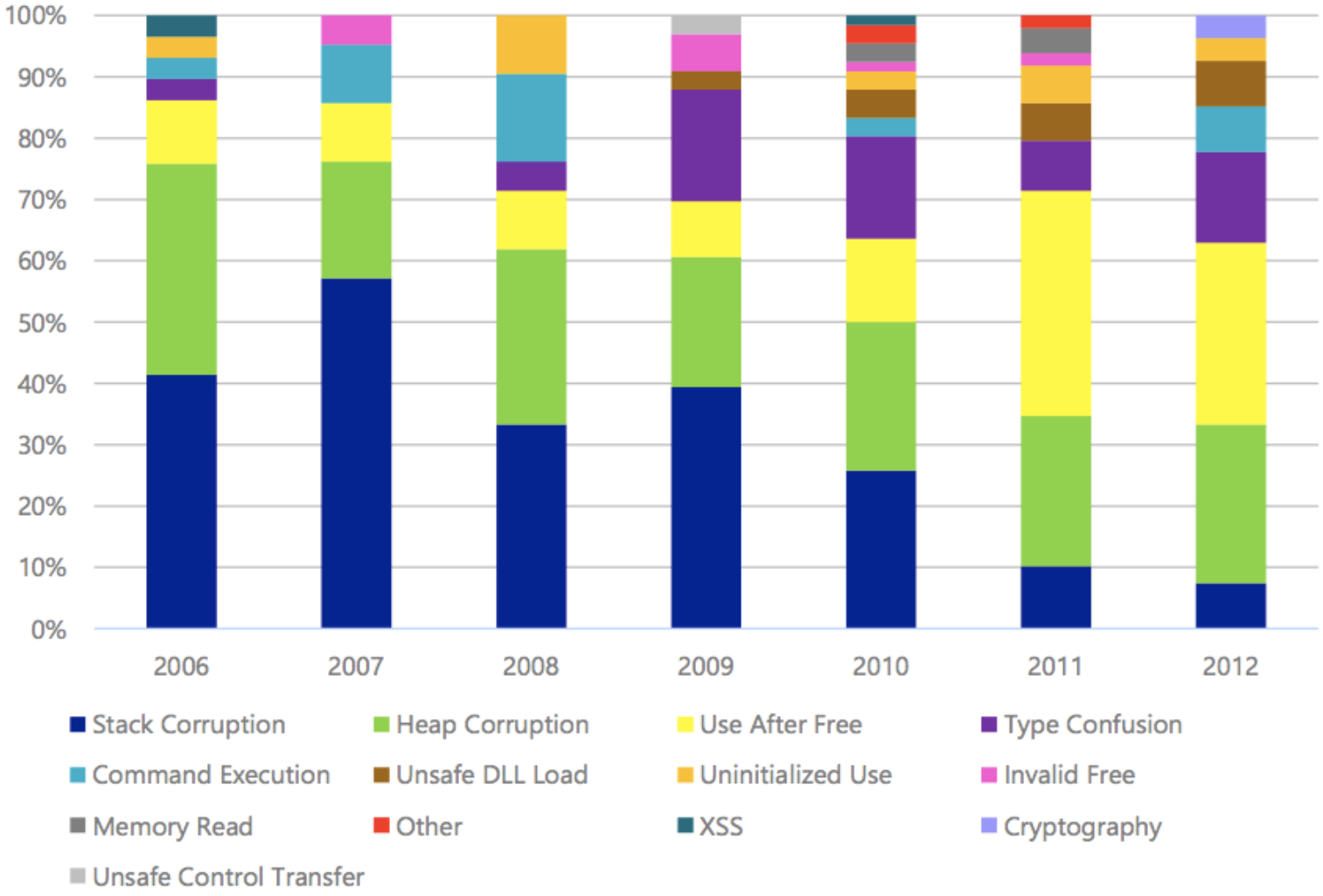


Figure 5. The distribution of CVE vulnerability classes for CVEs that are known to have been exploited

Real world.. Stuxnet

3 out of 4 bugs used were logic bugs

Exploit kits

Java.

TRAIL OF BITS

Practical

Not: How do I find use-after-free bugs?

But

How do I find a specific type of use-after-free bug in IE/Chrome?

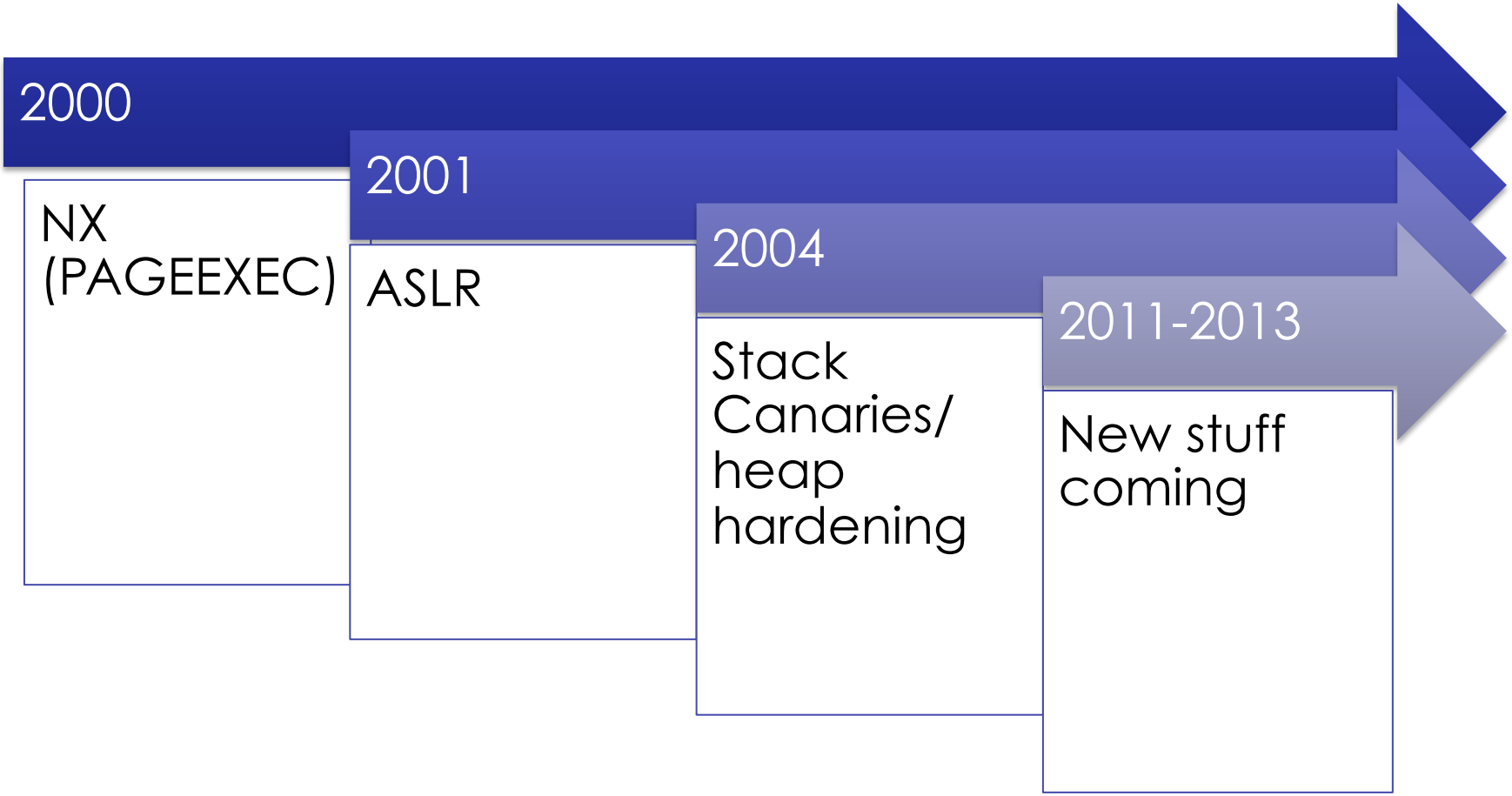
Future

Will automatic bug-hunting technique converge to AI in the future?

Can machines discover new bug patterns?

Exploitation

Quick recap(generic mitigations)



TRAIL OF BITS

Fact

Writing an exploit in 2013 is **theoretically** no different than writing one in 2005

Another fact

To date the lower bound on the number of bugs needed to compromise an application (sandboxes excluded) is almost always between 1 and 2

This shows

Exploitation becomes fundamentally application-specific above a certain number of kLOC

TRAIL OF BITS

Hot Stuff

	Policy type (main approach)	Technique	Perf. % (avg/max)	Dep.	Compatibility	Primary attack vectors
Generic prot.	Memory Safety	SofBound + CETS	116 / 300	×	Binary	—
		SoftBound	67 / 150	×	Binary	UAF
		Baggy Bounds Checking	60 / 127	×	—	UAF, sub-obj
	Data Integrity	WIT	10 / 25	×	Binary/Modularity	UAF, sub-obj, read corruption
	Data Space Randomization	DSR	15 / 30	×	Binary/Modularity	Information leak
	Data-flow Integrity	DFI	104 / 155	×	Binary/Modularity	Approximation
CF-Hijack prot.	Code Integrity	Page permissions (R)	0 / 0	✓	JIT compilation	Code reuse or code injection
	Non-executable Data	Page permissions (X)	0 / 0	✓	JIT compilation	Code reuse
	Address Space Randomization	ASLR	0 / 0	✓	Relocatable code	Information leak
		ASLR (PIE on 32 bit)	10 / 26	×	Relocatable code	Information leak
	Control-flow Integrity	Stack cookies	0 / 5	✓	—	Direct overwrite
		Shadow stack	5 / 12	×	Exceptions	Corrupt function pointer
		WIT	10 / 25	×	Binary/Modularity	Approximation
		Abadi CFI	16 / 45	×	Binary/Modularity	Weak return policy
		Abadi CFI (w/ shadow stack)	21 / 56	×	Binary/Modularity	Approximation

"Eternal War in Memory"
Laszlo Szekeres, Mathias Payer, Tao Wei, Dawn Song

TRAIL OF BITS

Again, get specific.. CFI

Vtguard/Vtable protection in Chrome

SEHOP/ Stack canaries

EMET

Some more, memory safety

DOM Objects “heap partition”:
https://code.google.com/p/chromium/issues/detail?id=246860

LFH allocation order randomization

UDEREF (PaX)

TRAIL OF BITS

Future

Adaptive exploits/Probabilistic exploits

Data-only exploits

AEG?

TRAIL OF BITS

Adaptive exploits

Information leaks become more and more important

Timing attacks become relevant as well (i.e. Dion Blazakis and pakt "Leaking addresses with vulnerabilities that can't read good")

Timing attacks

Problem: what can we tell about the program and heap states before we perform tasks that can crash the application?

Timing attacks - defense

Can we make operations on data structures have the same best, worst and average case complexity?

And heap allocators?

How about Garbage collectors with no noticeable slowdowns?

etc etc

Data-only attacks

Given a program state p and a memory corruption bug what data can I change to reach a ‘privileged’ state s ?

Note: solving this problem also helps a lot in solving its dual

Conclusion

GS Elevator Gossip
@GSElevator

Following

#1: When you tell a story, all I can think about is how much shorter it should be.

TRAIL OF BITS

First

Look for real-world data

Do your own offensive research

Develop intuition through practice

Each large size application is a research topic of its own (sad but true)

Second

Seek collaboration with the industry

Collaborate in funded projects (EU FP, DARPA CGC, etc.) with industry researchers

Integrate industry-led research in your curricula

Thanks!
Questions?

vincenzo@trailofbits.com