Repeated vs. single-round games in security

Halvar Flake / Thomas Dullien
Researcher at Google Project Zero

BSides Zurich 2017

Keynotes are the Golf ...

… of retired researchers.

- Today, economical thought is pervasive in information security

- Almost all discussions involve economic arguments:
  - “Increase the cost for the attacker”
  - “What is the trade-off of the cost for the defender vs. the cost of the attacker?”

Thesis of this keynote:

- We often fail to take the iterated nature of security into account
- This leads to suboptimal decisionmaking

Defender goals are often phrased as:

“Increase the cost for an attacker to …

… find an exploitable bug in X

… exploit a bug in X

… compromise organisation X

What we should be thinking about is similar to “marginal cost”:

What does it cost to …

… find an additional bug n+1 in software X if you have already found n bugs ?

What we should be thinking about is similar to “marginal cost”:

What does it cost to …

… exploit an additional bug n+1 in software X if you have already exploited n ?

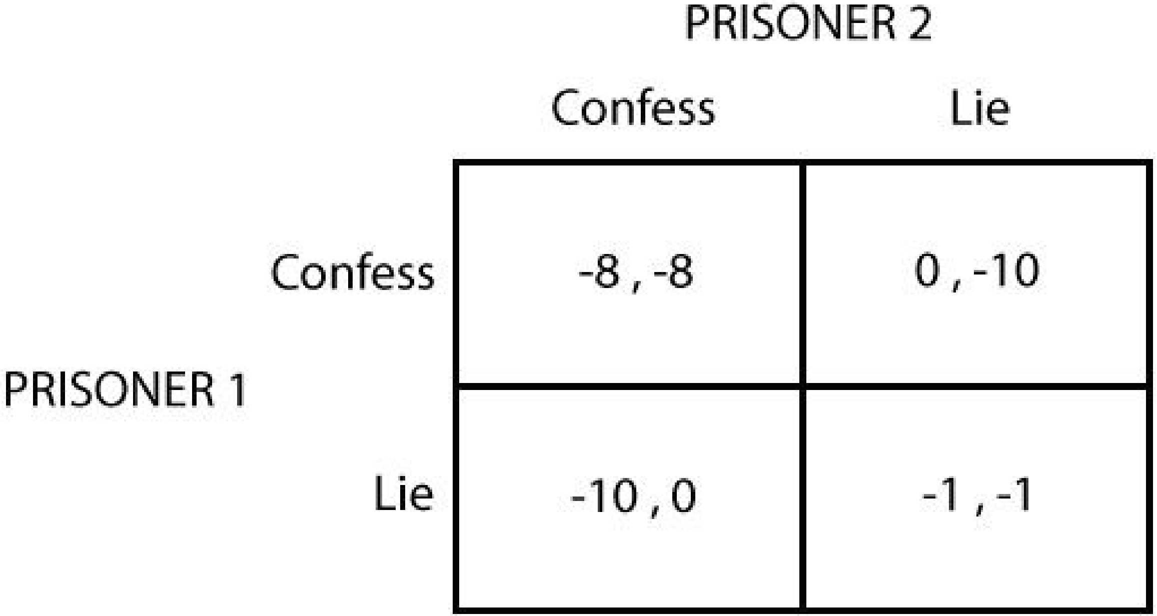
What we should be thinking about is similar to "marginal cost":

What does it cost to …

… compromise an additional organisation n+1 if you have already compromised organisation n?

Why should we think about these things?

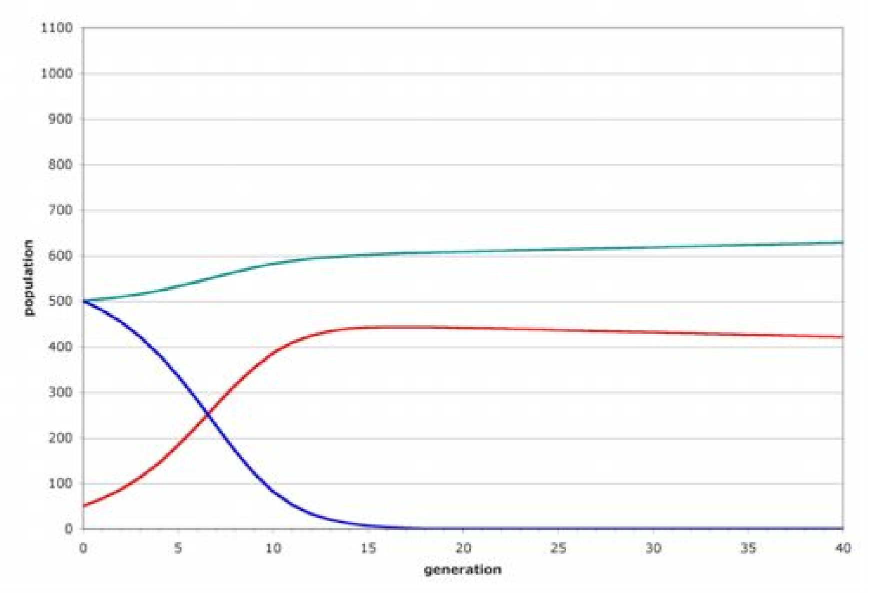
Famous prisoner’s dilemma:

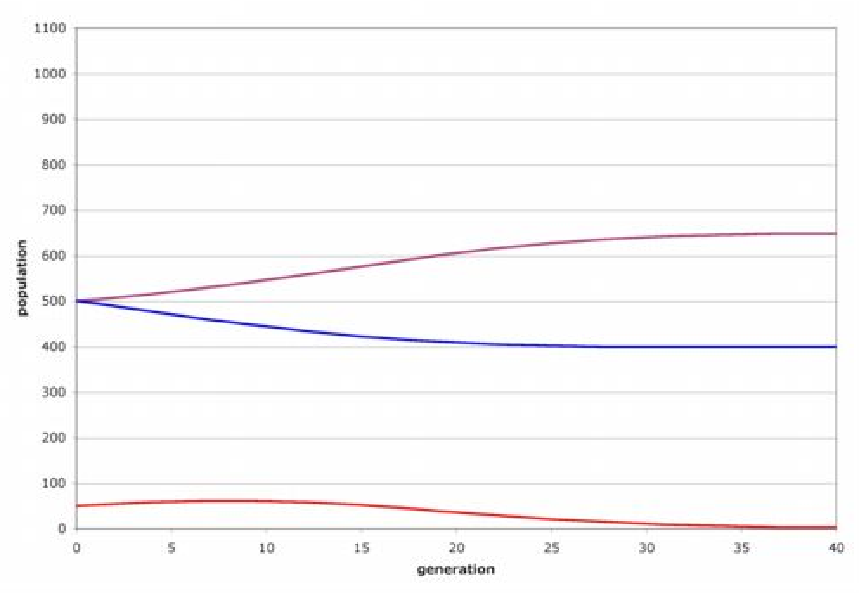


		PRISONER 2	
		Confess	Lie
PRISONER 1	Confess	-8 , -8	0 , -10
	Lie	-10 , 0	-1 , -1

Why should we think about these things?

Societal iterated prisoner’s dilemma





Important lesson:

Games change their dynamics drastically when they are played over multiple rounds.

Cost calculous changes drastically when repetition enters the picture.

Outline of the talk

1) Discuss how the costs for attackers change if we consider repetition for:
    a) Finding security vulnerabilities / bugs in a given target
    b) Writing exploits for a given target
    c) Compromising an organisation
2) Discuss some examples of suboptimal results arising from this.

Marginal cost of vuln-hunting in modern browsers

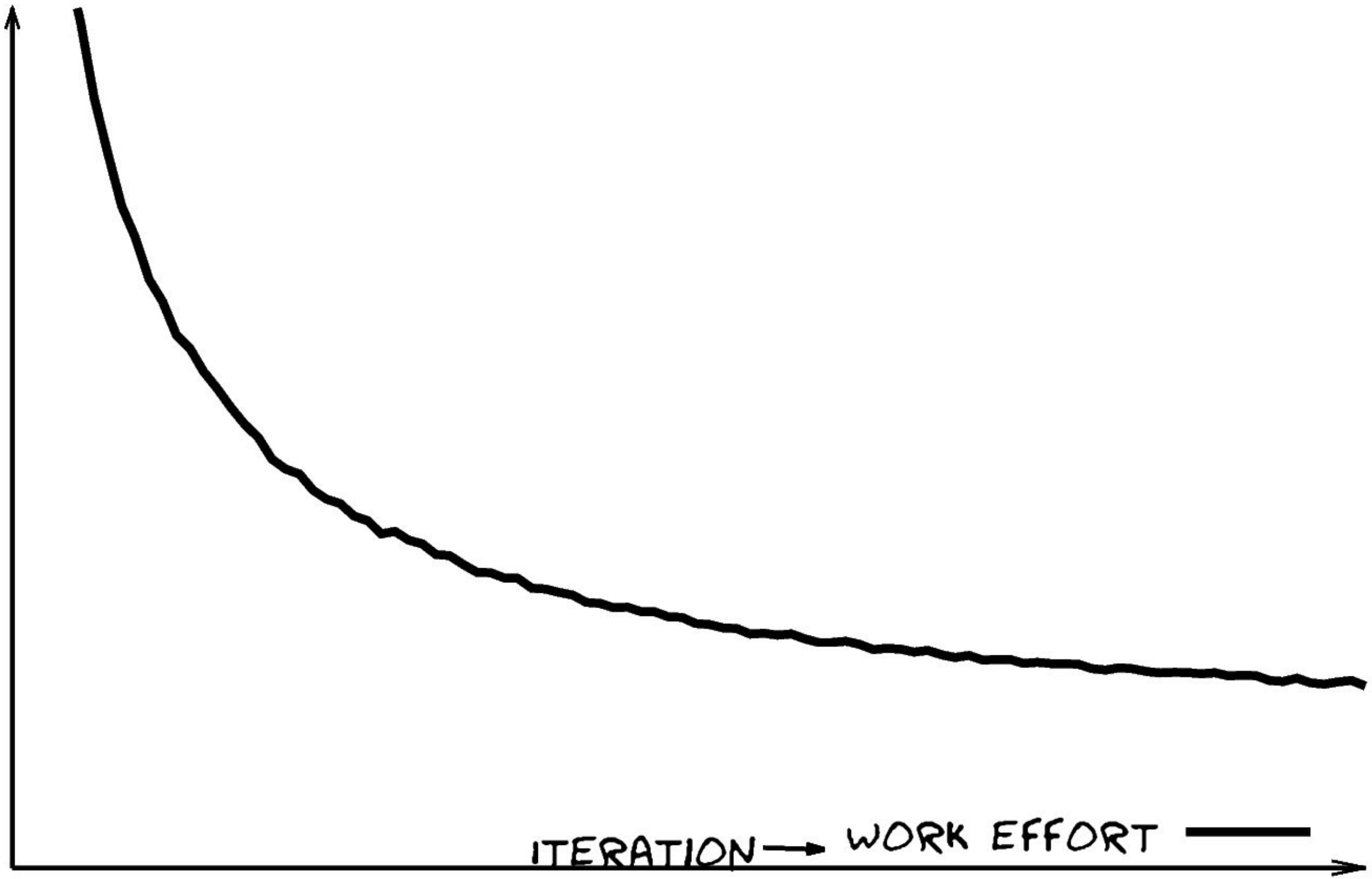
- Massive ramp-up costs in a very large codebase
- Somewhere between 1 and 4 months to really get into it
- Bugs are not scarce, though!

Note about all plots that follow

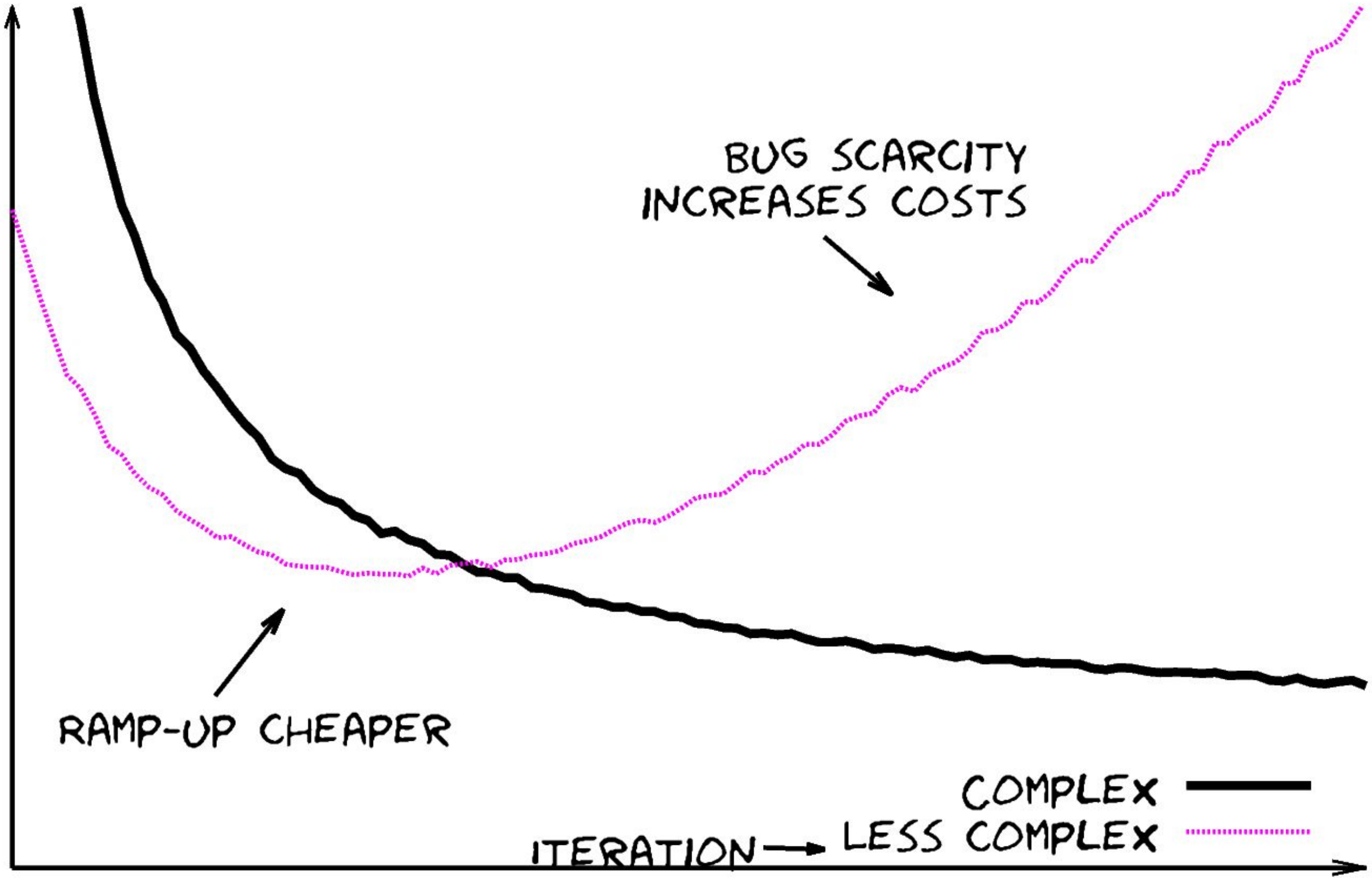
- Plots are not exact, only to illustrate concepts

- When the plot says “cost”, it means “expectation value of cost” - true costs for vuln-dev are randomly distributed around that expectation value

COST OF FINDING THE NEXT BROWSER BUG



COST OF FINDING THE NEXT BUG



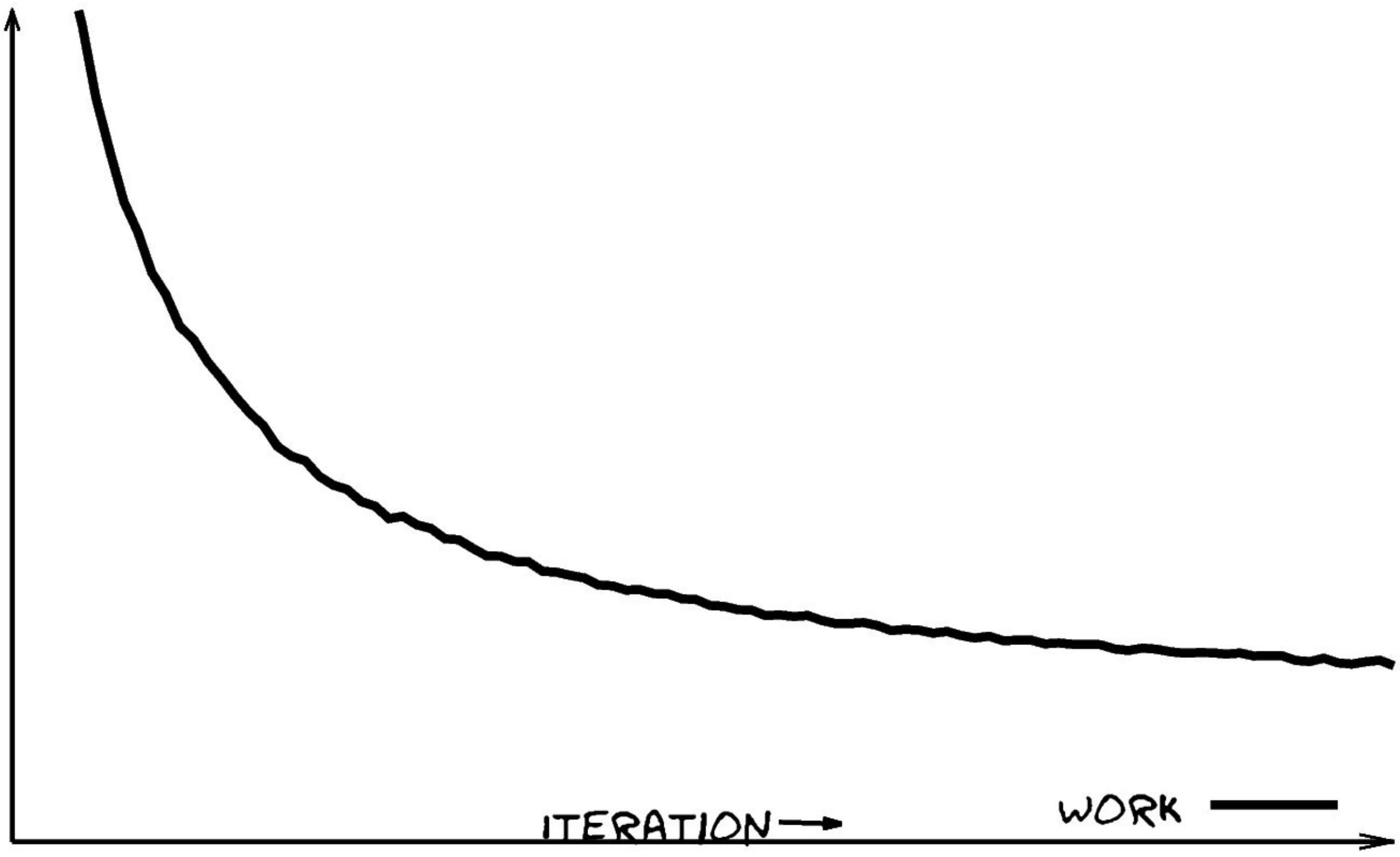
Result: False sense of security

“It took researcher X more than N months to find a critical vulnerability, so the cost of doing so is greater or equal to N months”

How about exploit development?

- Bug leads to the emergence of a “weird machine”
- Attacker needs to learn how to control & program that “weird machine”
- “Weird machine” is emergent property of target AND the bug in question
- Similar bugs in the same target yield similar weird machines, though!
- Similar bugs in similar targets can still yield slightly similar weird machines (browsers!)

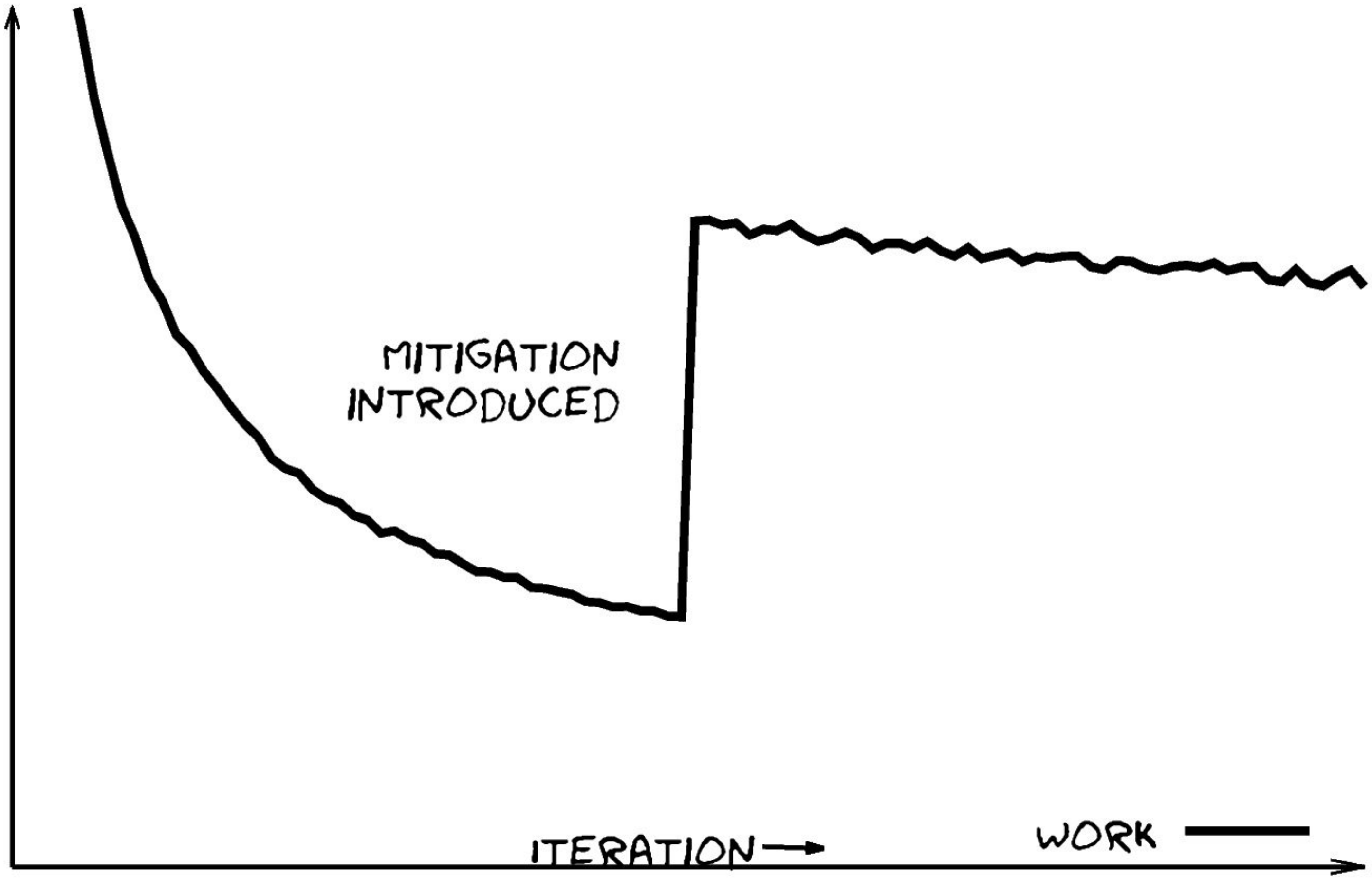
EXPECTED COST OF ITERATED EXPLOIT DEVELOPMENT GIVEN BUG IN FIXED TARGET



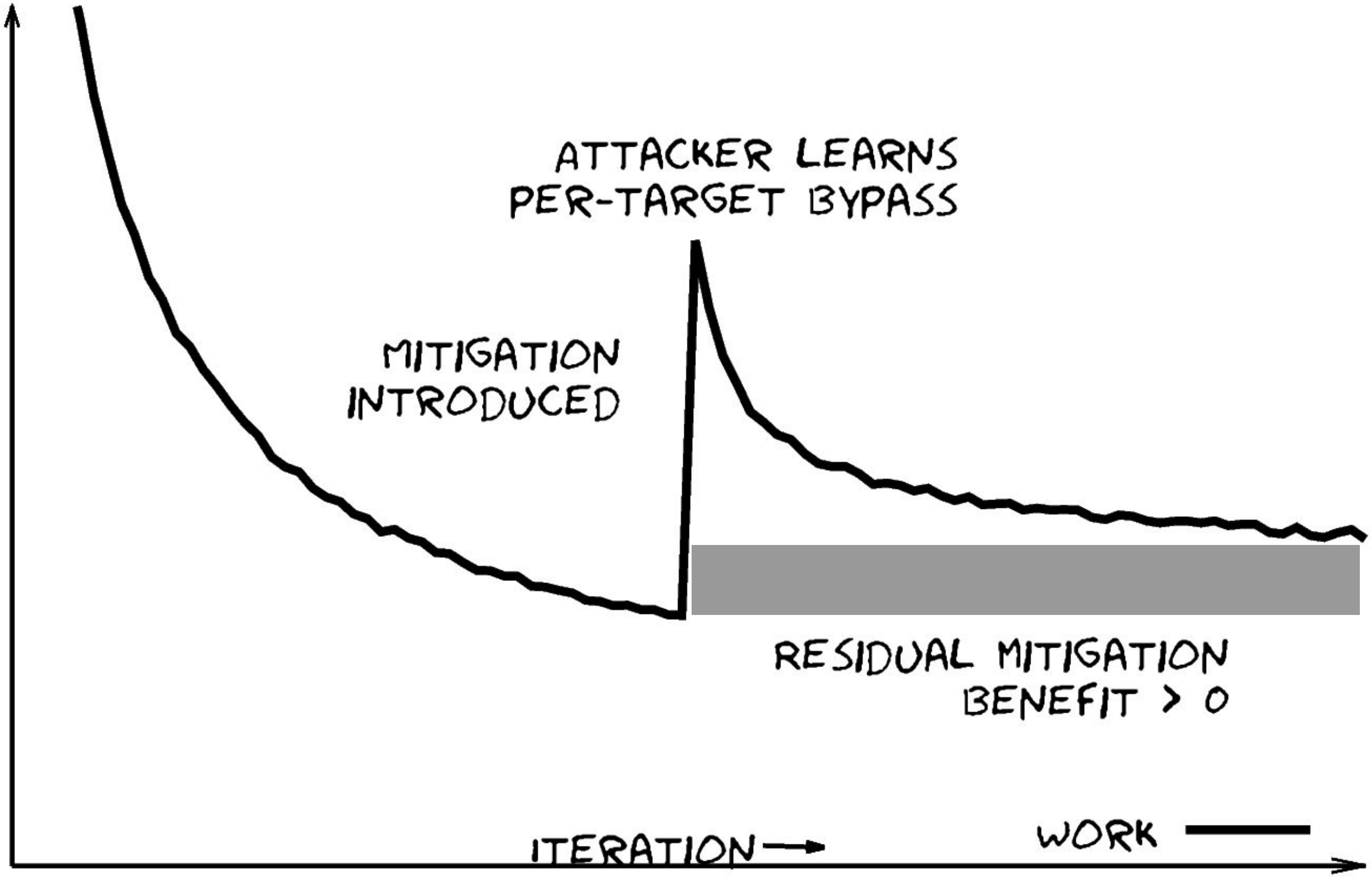
Falling costs affect mitigations, too

- Breaking ASLR is a cost paid per application + bug class, not necessarily per bug
- Breaking DEP is a cost paid per application + bug class, not necessarily per bug
- … etc etc etc ...

WHAT PEOPLE THINK THE EFFECTS OF MITIGATIONS ARE



WHAT PEOPLE THINK THE EFFECTS OF MITIGATIONS ARE



Result: False estimate of cost / benefit

- Most mitigations have complexity cost
- Most mitigations have inspectability / debuggability cost
- Both are paid in perpetuity by defenders and legitimate users
- Trade-off evaluation is often made myopic, at point-in-time:
    - Is today’s cost of the mitigation manageable for the defender / user & does it create cost for the attacker now?

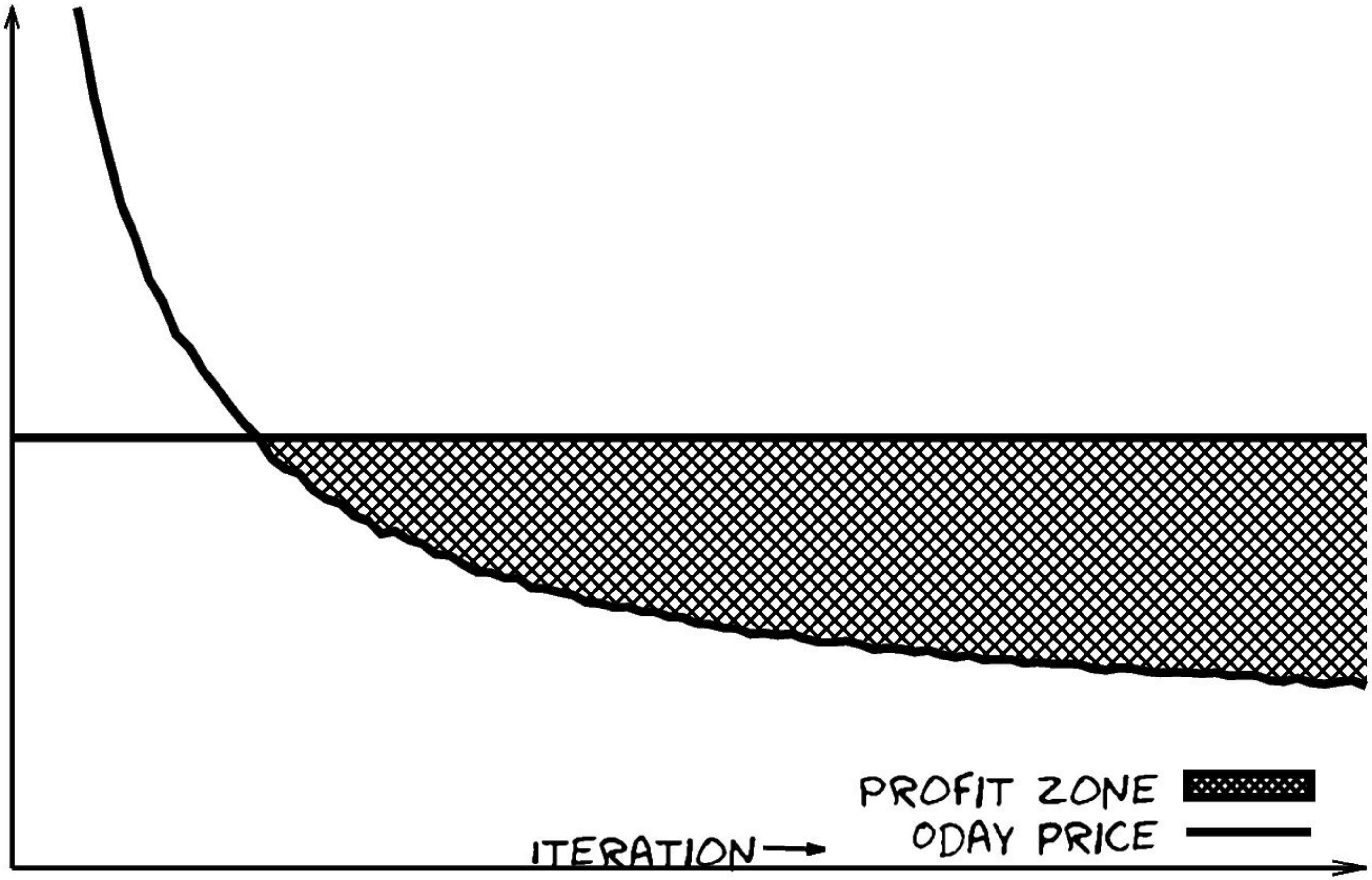
Better way of evaluating cost / benefit

- We need to integrate over time.

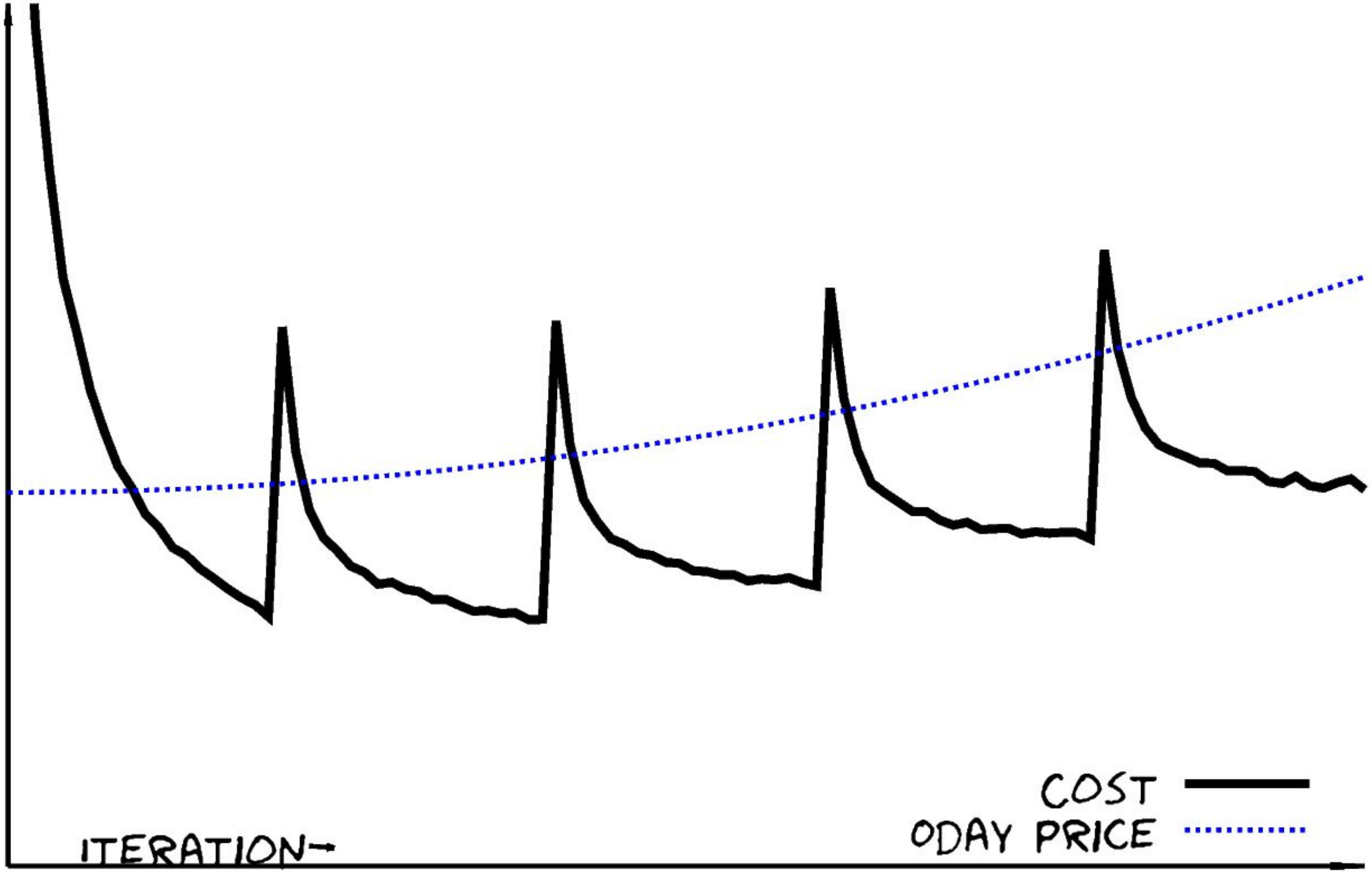
- Is the sum of all future costs from this mitigation, in perpetuity, worth the expected long-term residual benefit of the mitigation?

Digression: 0day vendor business model

0DAY VENDOR BUSINESS MODEL



MORE REALISTIC 0DAY VENDOR BUSINESS MODEL



Questions

- 0day prices will keep rising at the same rate that digitization proceeds - can we really bend the curve upward using only mitigations to catch up?
- Right side of cost curve steepens when software complexity is low and bugs get scarce. Some software projects have an exponentially costly right side (OpenSSH etc.) - but no perpetually-buggy software has been mitigated into security.

Development of the last 15 years

- We have bent the **left side** of the cost curve up steeply

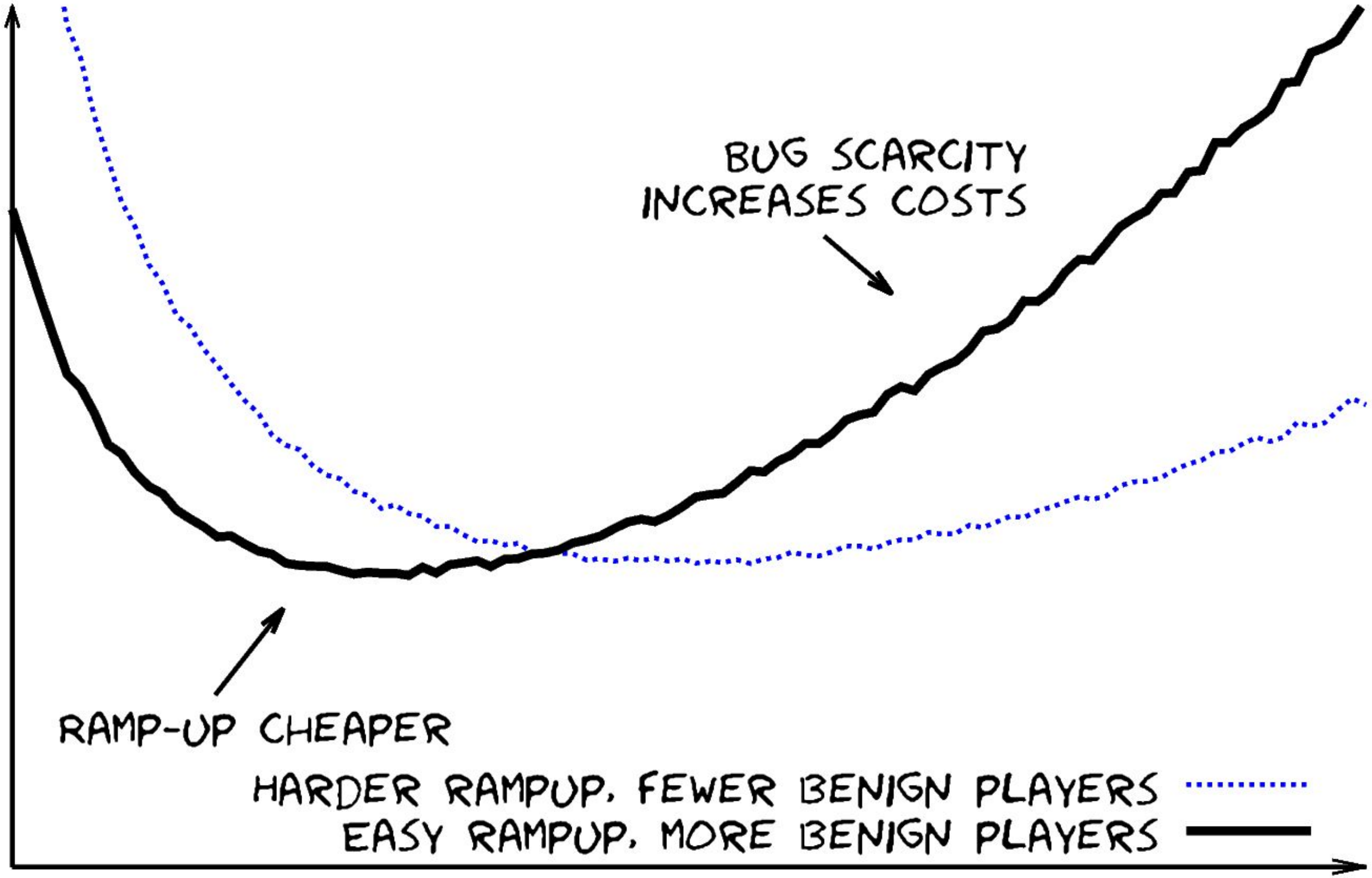
- Ramp-up is much more expensive now

- Higher software complexity + various mitigations

Unintended side-effects of this

- Making ramp-up harder primarily eliminates benign players
- No hobbyists, fewer people doing it for the “fun”
- Very few people are willing to invest 6+ months full-time into a hobby project w/o payout

EFFECT OF HARDER RAMP-UP



BUG SCARCITY INCREASES COSTS

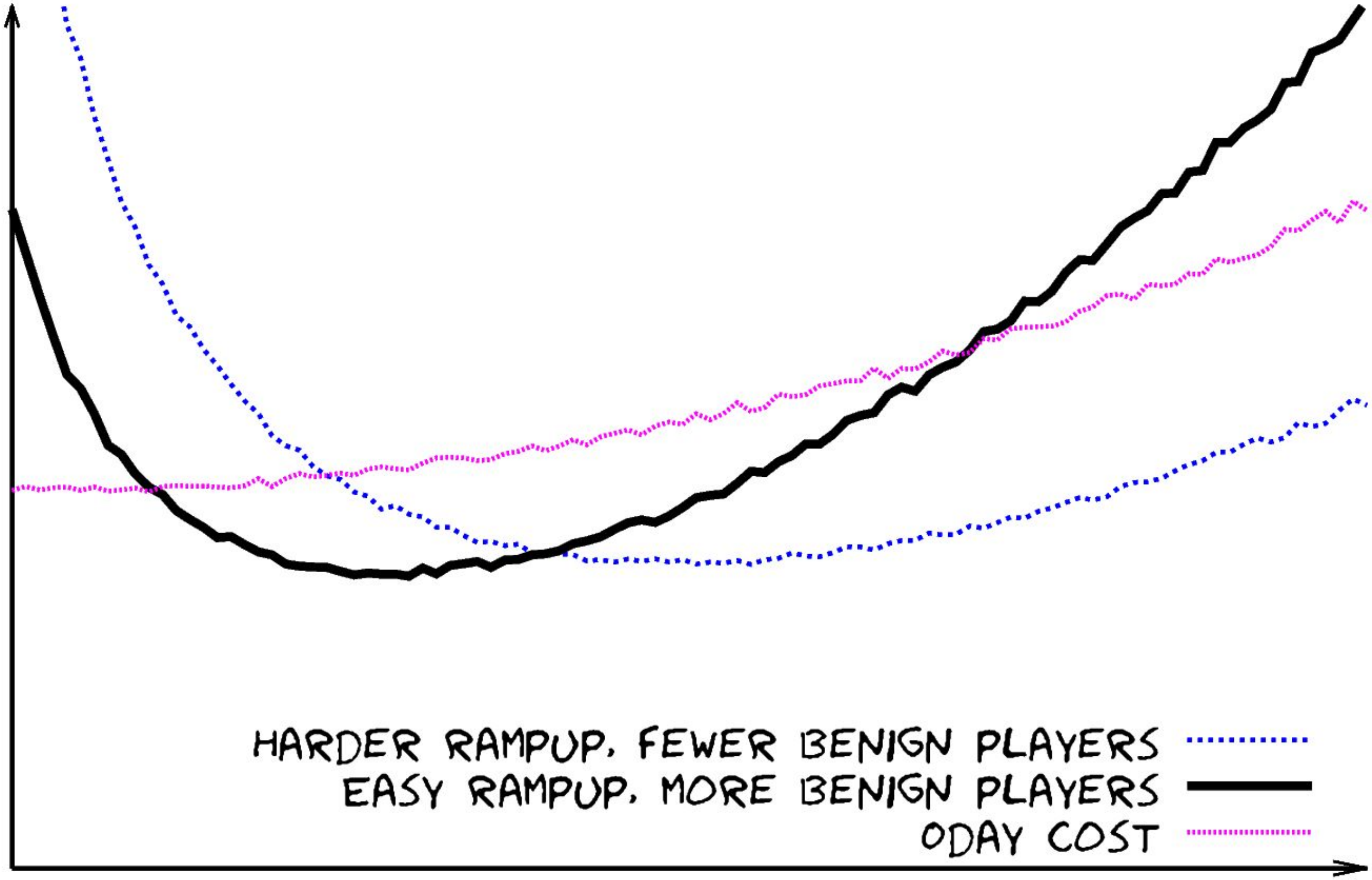
RAMP-UP CHEAPER

Unintended side-effects of this

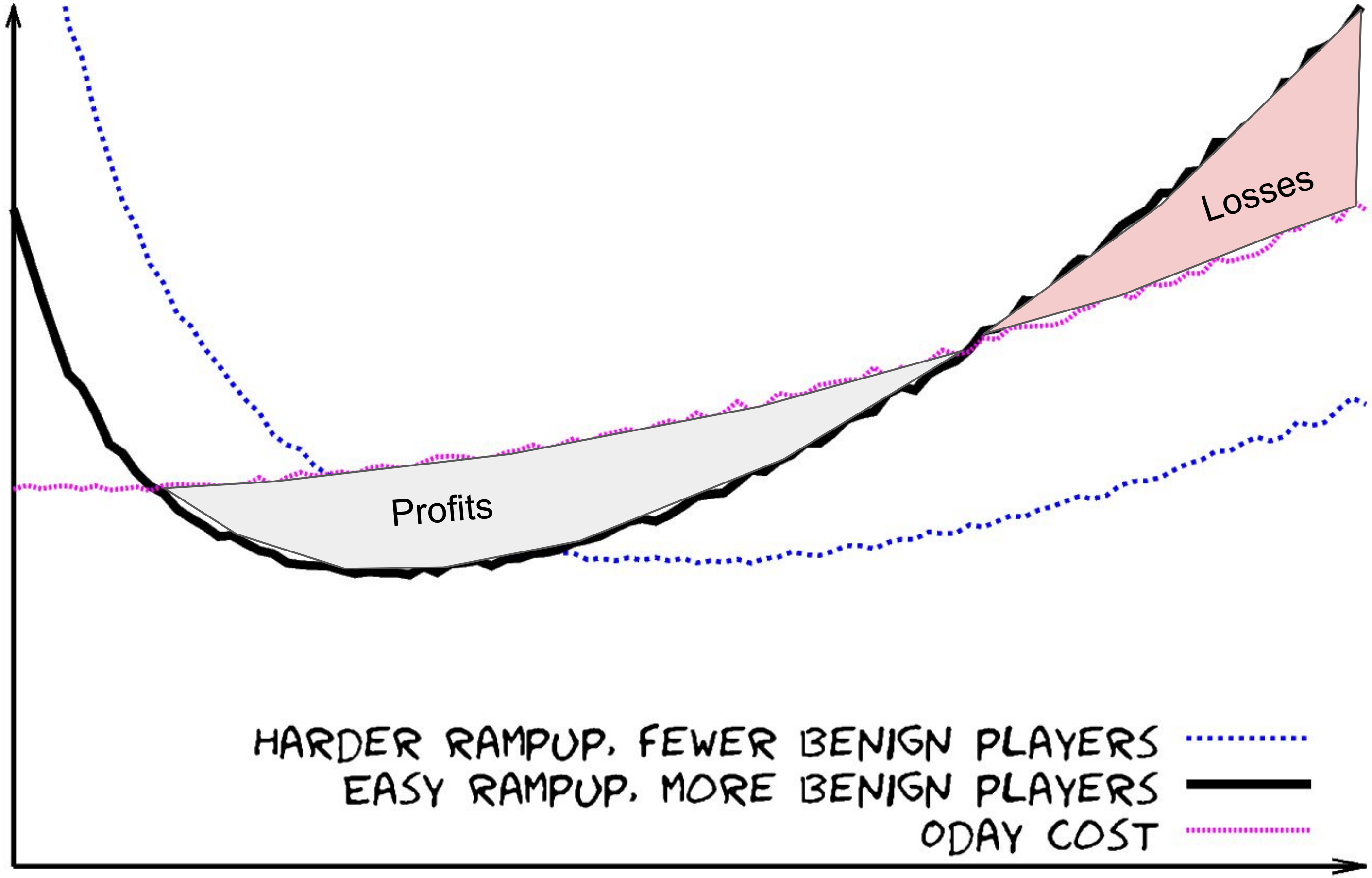
Removing benign players **flattens** the curve to the right (as upward slope from bug scarcity gets flattened!).

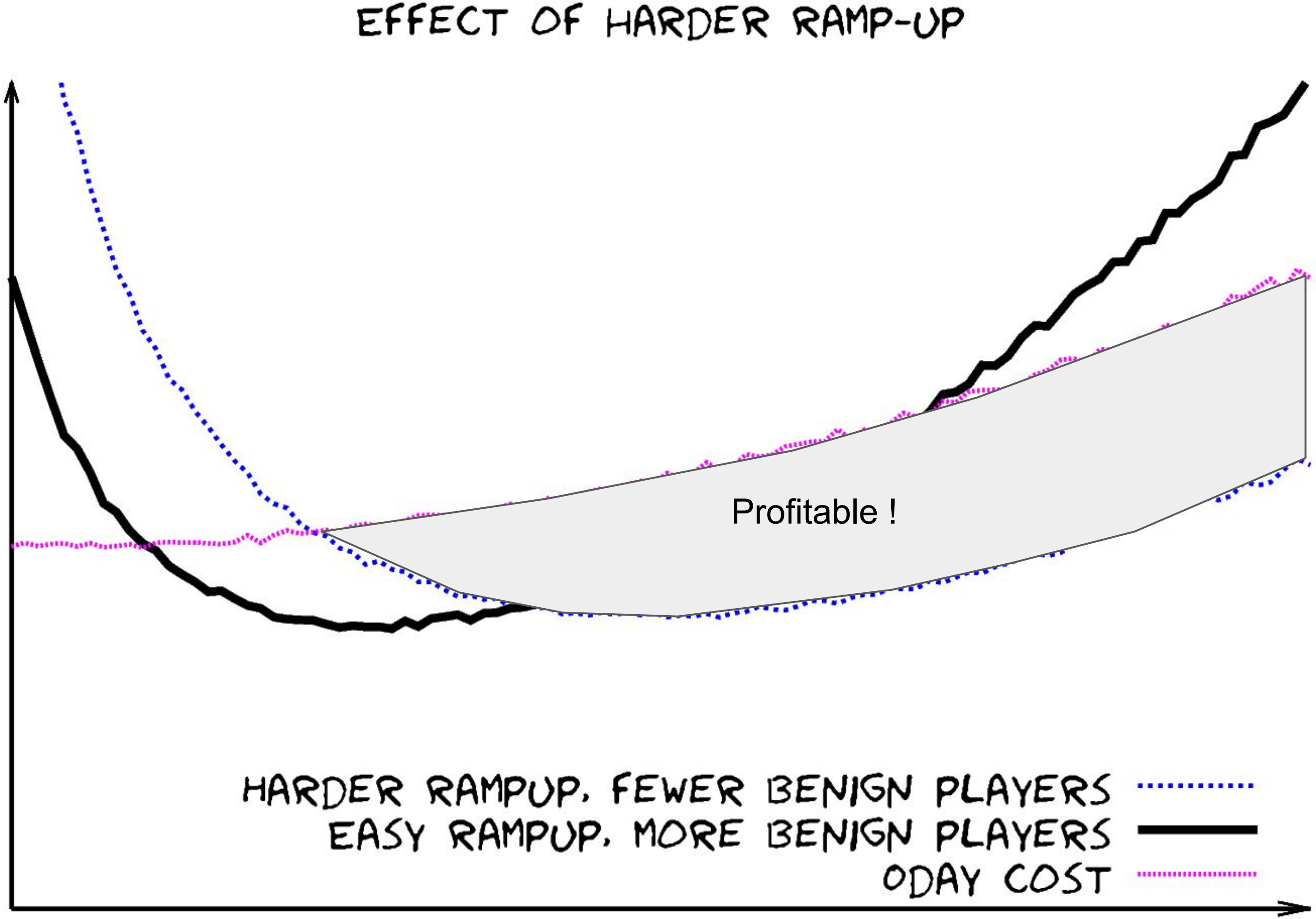
It is quite possible that we have improved the long-term economics of the 0day vendors by making “getting started” hard.

EFFECT OF HARDER RAMP-UP



EFFECT OF HARDER RAMP-UP





EFFECT OF HARDER RAMP-UP

Example: Harder debugging on most platforms

- Only platform where debugging is better in 2017 than in 2007 is Linux
- All other platforms have gotten harder to debug, harder to introspect etc.

Repeat attackers pay the price for proper debugging and introspection only once.

Example: MPEngine Lockdown

- “Protected Processes” - Windows programs that you cannot debug with a usermode debugger, even if you have all privileges
- Attackers can load a signed vulnerable driver, run an exploit, get execution & deprotect the process - so … why?

Example: Locked-down platforms

- In order to do meaningful research on a modern phone, you need to have local root exploits
- As defender, you are not supposed to hoard 0day, right?

Example: Locked-down platforms

- Defenders have to pay a constant tax (in the form of finding local roots, writing exploits for them, reporting them, and cycle) to perform **defensive research**.
- Attackers can take a sub-par / low-reliability bug they have anyhow, keep it forever and save that tax.

Some security measures have become like DRM:

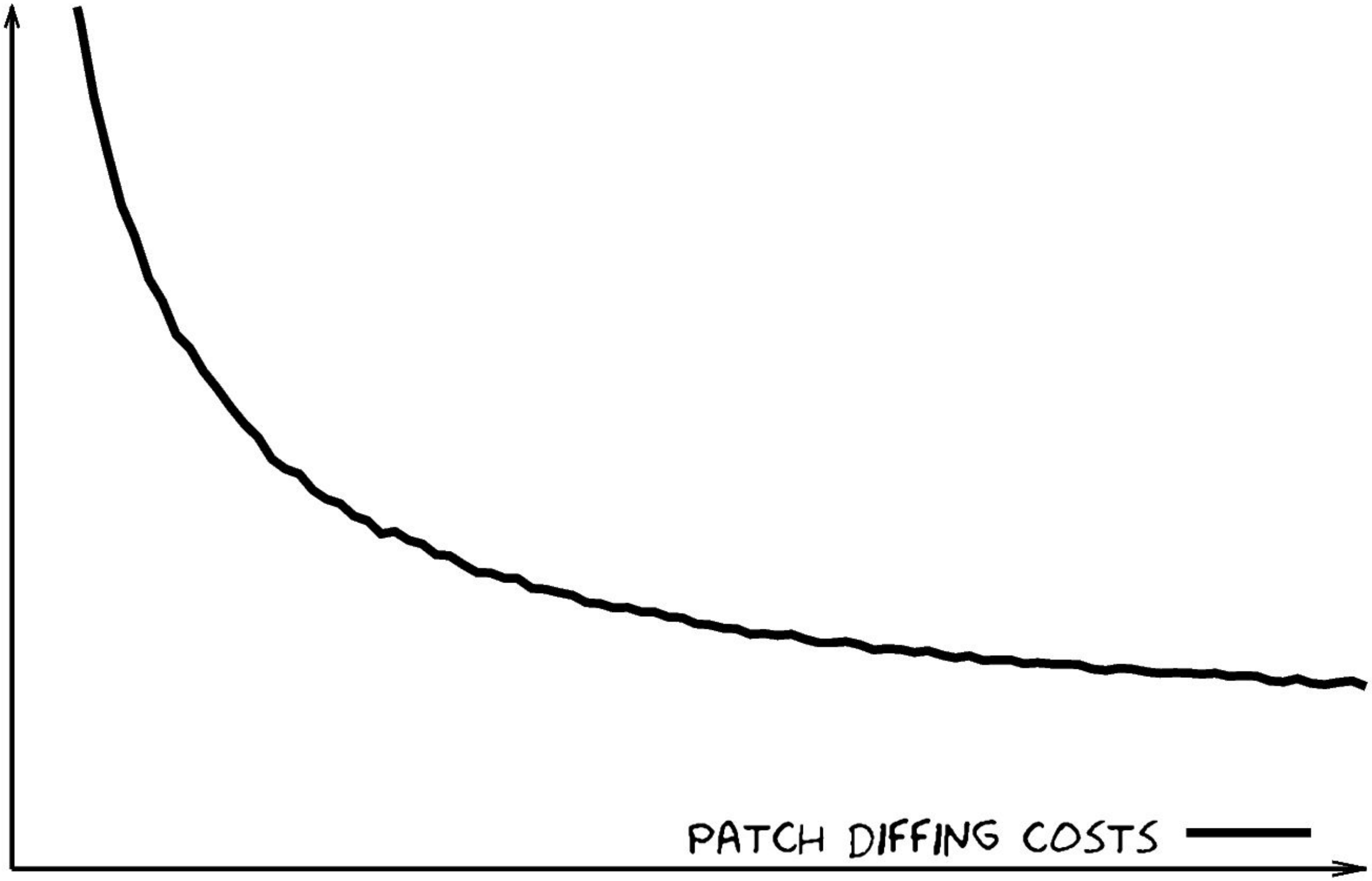
They primarily inconvenience the good guys.

How about patch diffing?

- Shipping binary patches for bugs and acting like they are not effectively disclosed to anyone that cares
- “Zombie idea” - extremely stupid and impossible to kill
- Gets hit with a hammer each year since 2004, still shuffles on

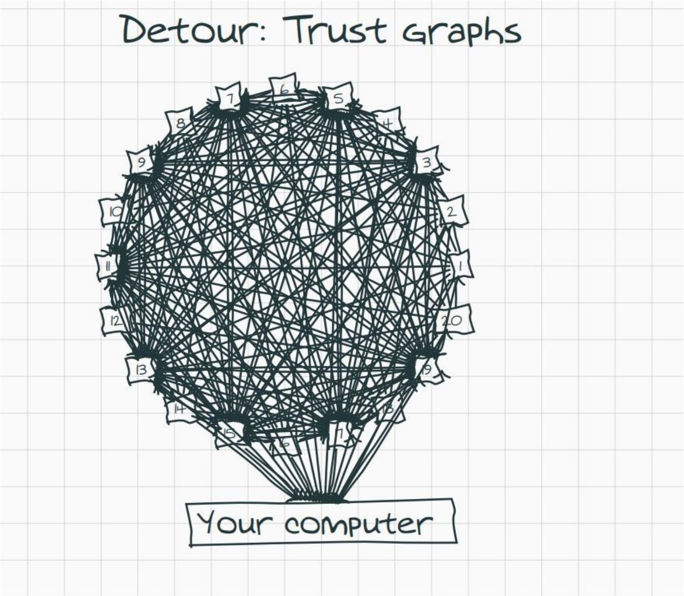
- First-time analyzing a patch in a given target: Hard
- N-th time analyzing a patch in a given target: Easy

COST OF PATCH DIFFING FOR A TARGET



Compromising company n+1

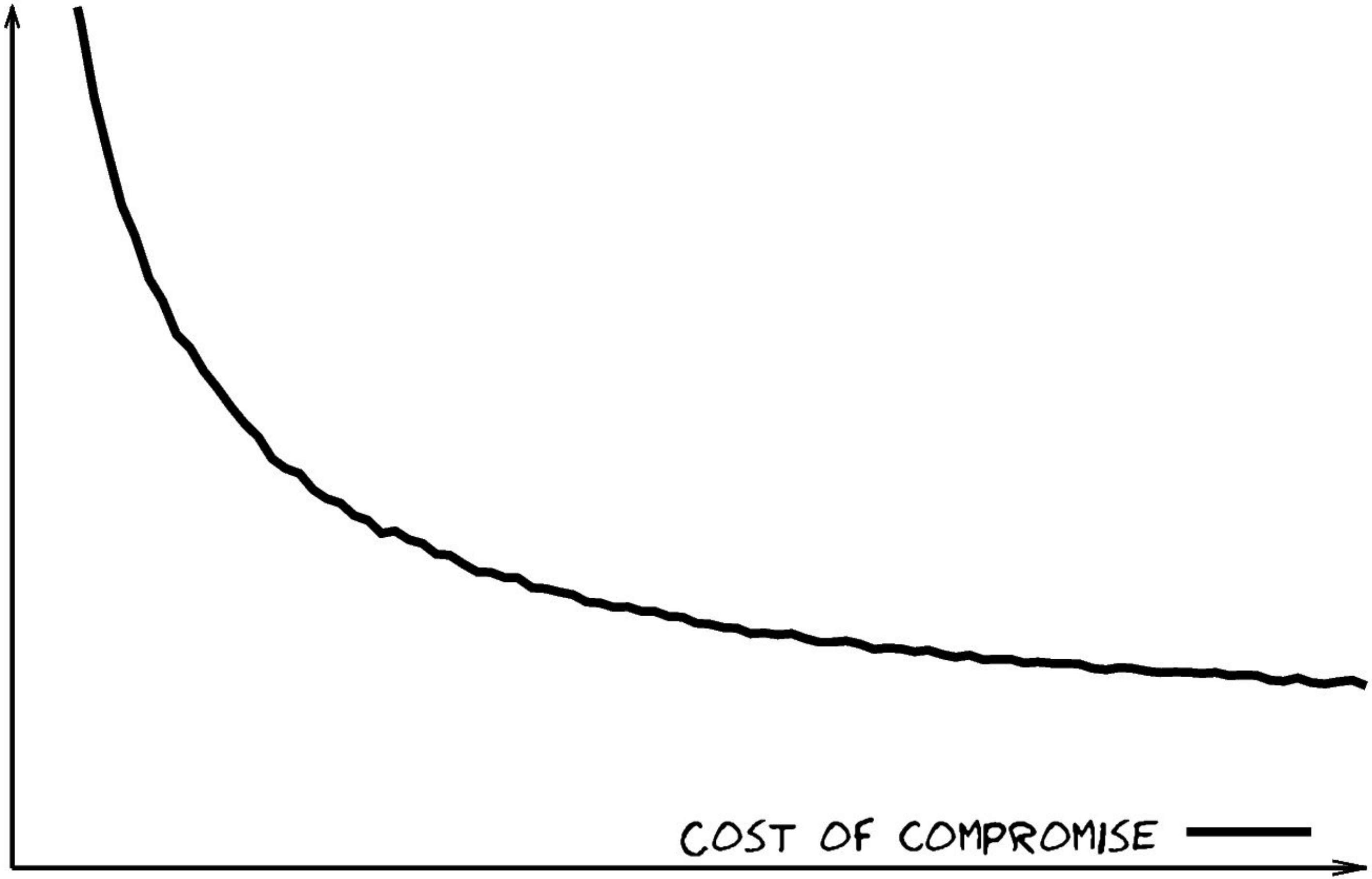
- IT is extremely tightly connected via trust relationships
- Trust is transitive
- Everybody is only one step away from everybody



Compromise boundary

- In a transitive trust graph, the number of nodes you can compromise at near-zero cost grows exponentially with the number of nodes you control
- It is rarely “how expensive is it to compromise organisation X”, it is “how expensive is it to compromise organisation X if I have Y, Z, and K”
- Decreasing marginal costs for the attacker, again.

COST OF COMPROMISING COMPANY N+1 IF YOU HAVE 1 TO N



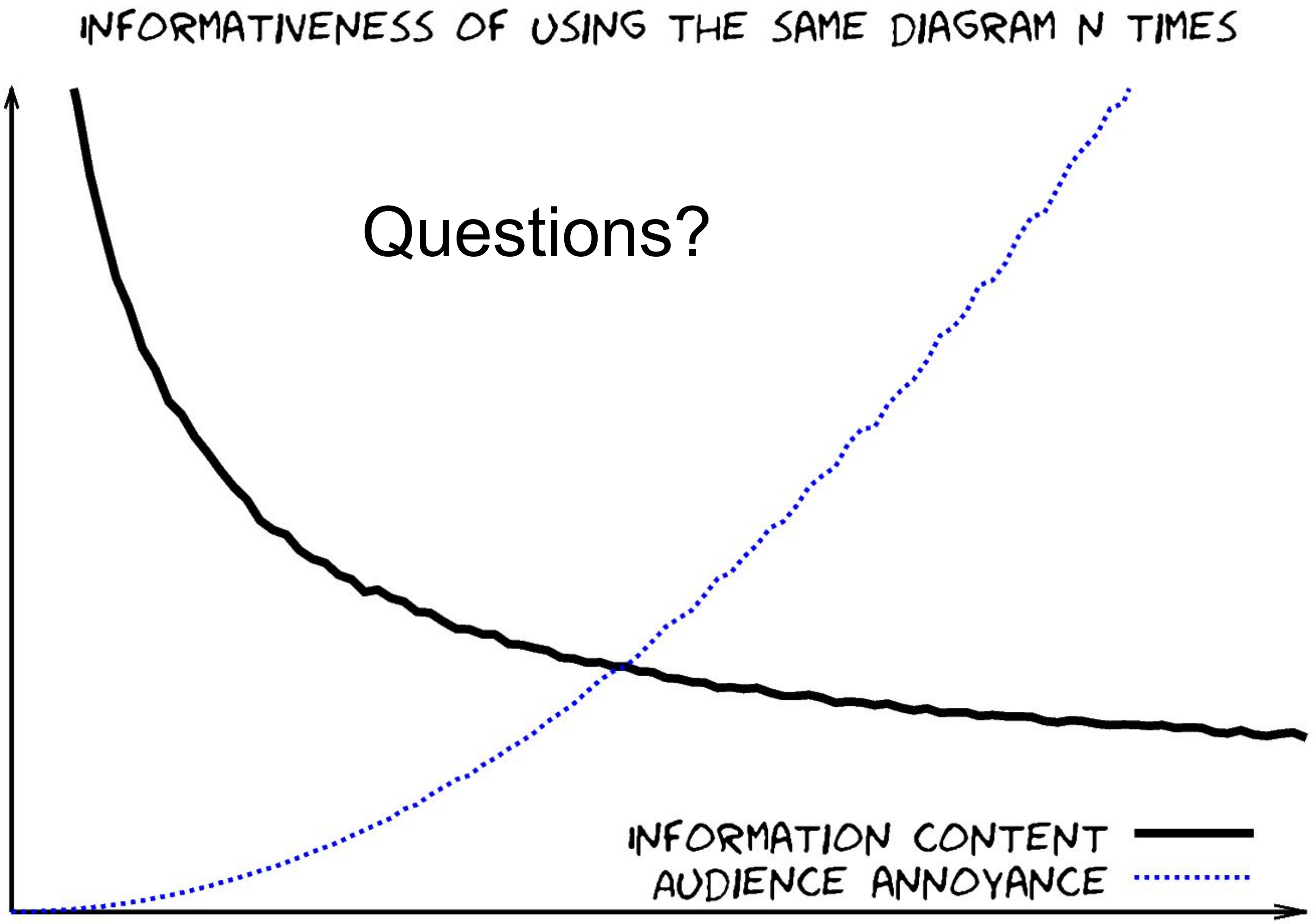
Transitive compromise is common

- BitchX IRC client 2002
- CCleaner 2017
- Custom Apache modules were common as early as 2002 to serve backdoored software only to specific targets

Did I mention that security is sometimes a bit repetitive?

Summary

- Security is full of repetition, and any relationship with an adversary is a repetitive game
- As an industry, we generally ignore the differences in marginal costs over the many repetitions
- Focus on single-shot costs has absurd side effects - encumbering benign researchers, potentially improving the long-term economics of 0day vendors, imposing ill-thought-out costs on users
- Understanding long-term marginal costs needs to be higher priority - it is hard to steer a car if you can only see 5 meters ahead


INFORMATIVENESS OF USING THE SAME DIAGRAM N TIMES
INFORMATION CONTENT
AUDIENCE ANNOYANCE

Questions?

Credits

- This talk grew out of long and complex discussions with my colleagues
- I would like to particularly thank Ian Beer, Mark Brand, Ivan Fratric, and Jann Horn for being fantastic sounding boards & critical reviewers of my thoughts