Lecture TU Darmstadt

From Science to Products – Security Research at Intel Labs

Matthias Schunter + slides from Steffen Schulz, Rafael Misozky, Jan Richter, …

intel®

Legal Information

The views and opinions expressed in this presentation are those of the author and do not necessarily represent official Intel policy or position.

No product or component can be absolutely secure. Your costs and results may vary.

These materials are provided "as is." Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

“In theory, theory and practice are the same. In practice, they are not.”

Outline

1. Security in practice – does anyone care?
2. Industrial vs. Basic Security Research
3. Example Research Projects
   a) Fuzzing Low-level Software
   b) Post Quantum Crypto
4. Discussion / Q&A

Case Study: Secure electronic cash anyone?

Excercise (via chat)

- *Advantages wrt Cash/CC?*
- *Disadvantages wrt Cash/CC?*

Cash → ? → Digital Cash in Digital Wallet

Credit Card → ?

Source: tellermate.com
eenewseurope.com
ECONOMICTIMES:COM

Case Study: Secure electronic cash anyone?

Cash

?

Digital Cash in Digital Wallet

Multiple Choice
- *Yes – this will work in practice*
- *No – this will not work in practice*

+why (into chat)

- Expensive & i

- e & Anonymous
- Efficient
- Remote value transfers

Credit Card

?

- Insecure; privacy-invasive

Source: tellermate.com
eenewseurope.com
ECONOMICTIMES:COM

Case Study: Secure electronic cash anyone?

Cash

?

Digital Cash in Digital Wallet

- Trusted anonymity

- Requires infrastructure
- Breaks habits
- Cost higher than insurance
- Interesting main usecase…

Credit Card

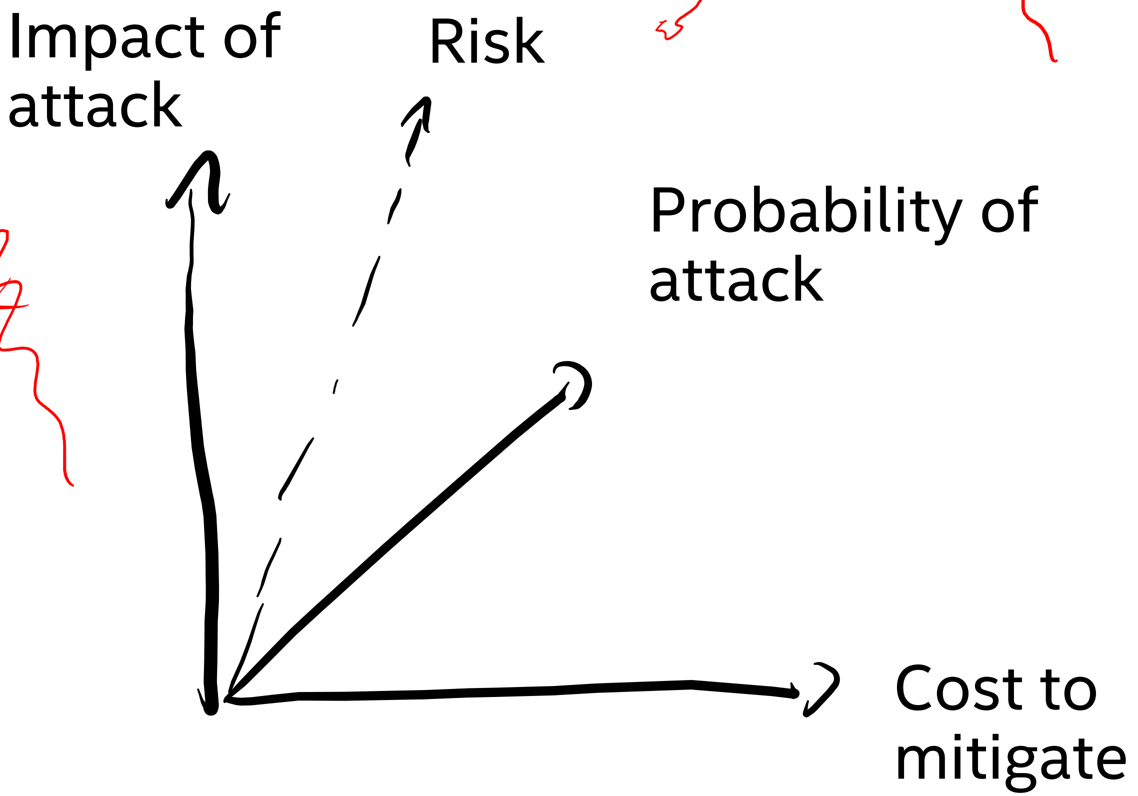
?

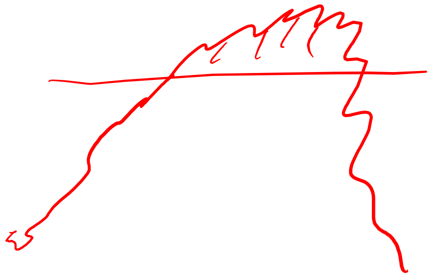
- Cheap insurance

Source: tellermate.com
eenewseurope.com
ECONOMICTIMES:COM

Is security important in the real world?

- Aim for a Rational Decision
  - Impact of attacks (cost+scale)
  - Probability of attacks
  - Cost to mitigate risk

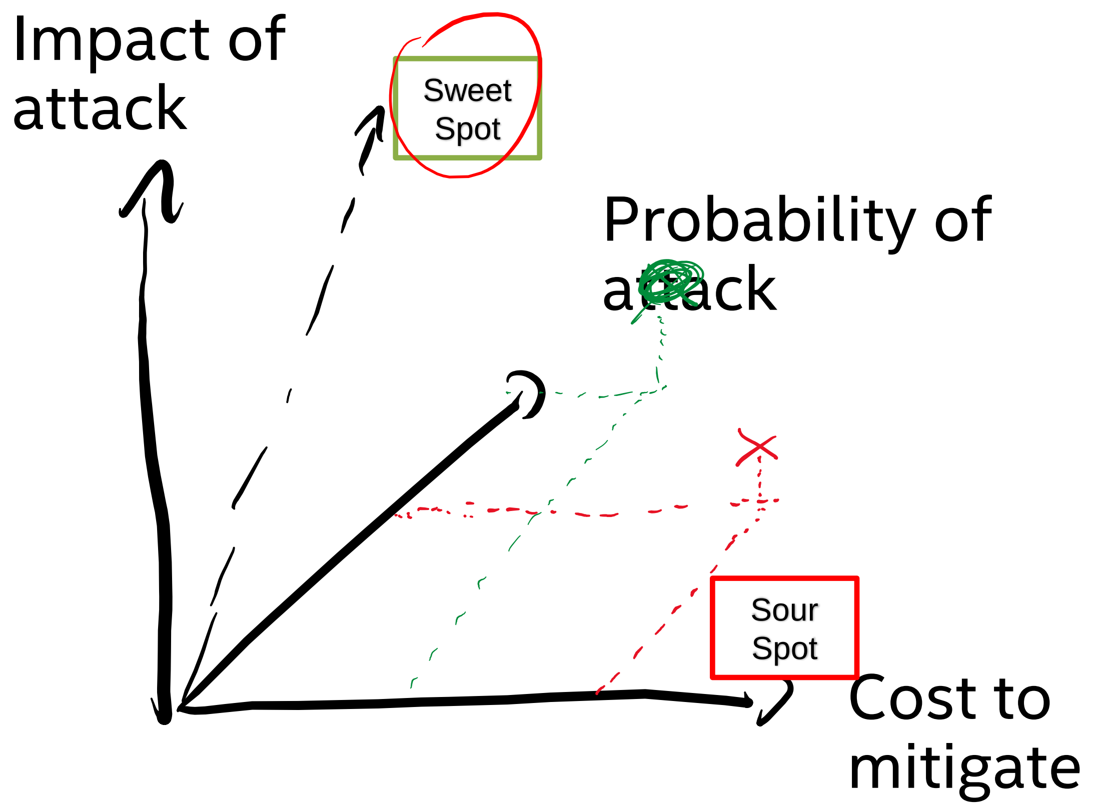




- But: Moving targets…

+ Incomplete Info.

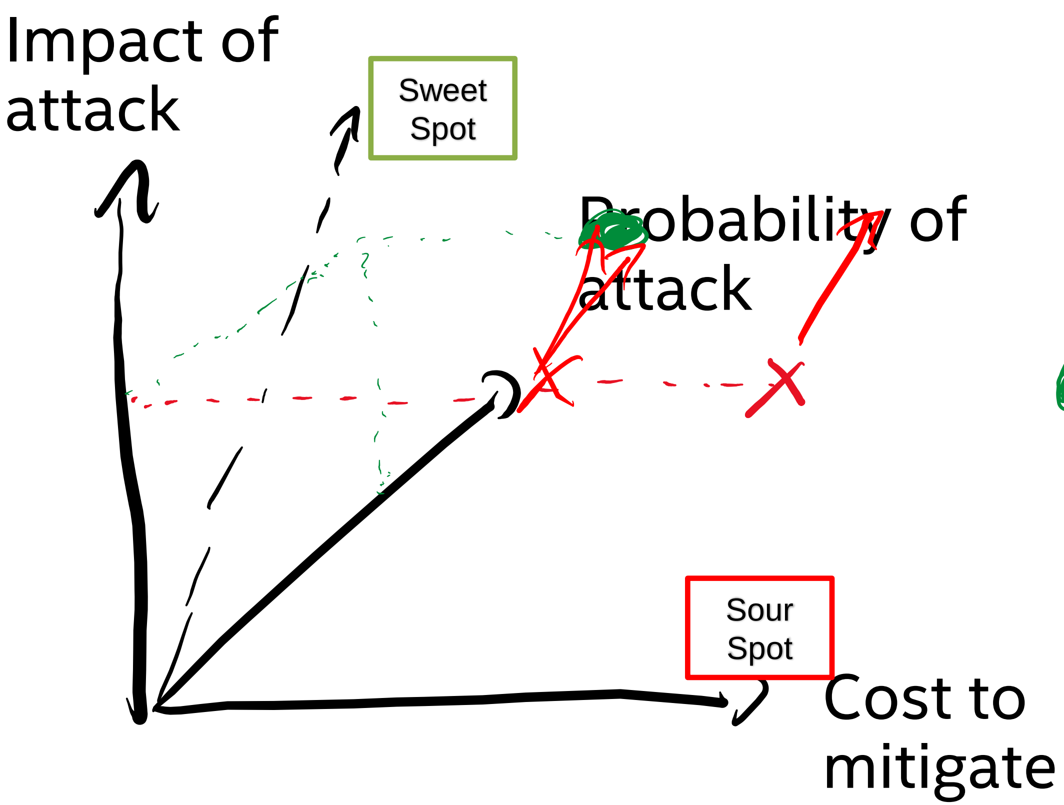
Is security important in the real world?



Security for Credit Cards

- 90s: No internet – limited scaling
  - Medium probability of attack
  - Limited cost per attack
  - High cost of mitigation (usability!)
- 2000s: Internet and ecommerce
  - High probability of attack
  - Limited cost
  - Medium cost of fixing

Is security important in the real world?



- Post quantum secure signatures
  - Today – no quantum computer
    - Zero probability of attack
    - High impact (*0 = zero risk)
    - High cost of mitigation
  - Once the QC machine has arrived
    - Medium probability of attack
    - High impact (high risk risk)
    - High cost of mitigation
  - But: Long life-time of HW

Security in Practice: Some lessons learnt

- Security is an attribute
  - Often engineered after the fact
    - Example: Security of neuro compute anyone?
  - Dynamically adjusted

- Security is a process
  - Secure Development Lifecycle
  - Includes response and repair!
    CVEs important for customer awareness!
  - Monitoring the environment is important!
    (90's PC on the Internet)
  - CVE counts have many causes and are no reliable metric across products/companies..

CVE Counts:

Product Name	Vendor Name	Product Type	Number of Vulnerabilities	
1	Debian Linux	Debian	OS	5078
2	Android	Google	OS	3651
3	Ubuntu Linux	Canonical	OS	2984
4	Mac Os X	Apple	OS	2759
5	Linux Kernel	Linux	OS	2668
6	Iphone Os	Apple	OS	2300
7	Windows 10	Microsoft	OS	2260
8	Chrome	Google	Application	2161
9	Windows Server 2016	Microsoft	OS	2021
10	Windows Server 2008	Microsoft	OS	1973
11	Fedora	Fedoraproject	OS	1959
12	Firefox	Mozilla	Application	1916
13	Windows 7	Microsoft	OS	1854
14	Windows Server 2012	Microsoft	OS	1728
15	Windows 8.1	Microsoft	OS	1636

Source: https://www.cvedetails.com/top-50-products.php

Security in Practice: Some lessons learnt

- Risk Management Decision
  - Risk = probability x impact
  - Mitigation cost
- Many moving targets…

Excercise (via chat or audio)
- *Any questions?*
- *Any feedback*

- Non-technical parameters are important!
  - Humans and training
  - Usage patterns and environment
  - All Employees play a role in security…
- Many important roles:
  - Crypto experts, SW security, SDL, …
  - Developers, users, …

Outline

1. Security in practice – does anyone care?
2. Industrial vs. Basic Security Research
3. Example Research Projects
   a) Fuzzing Low-level Software
   b) Post Quantum Crypto
4. Discussion / Q&A

Break...

Mission of Intel Labs

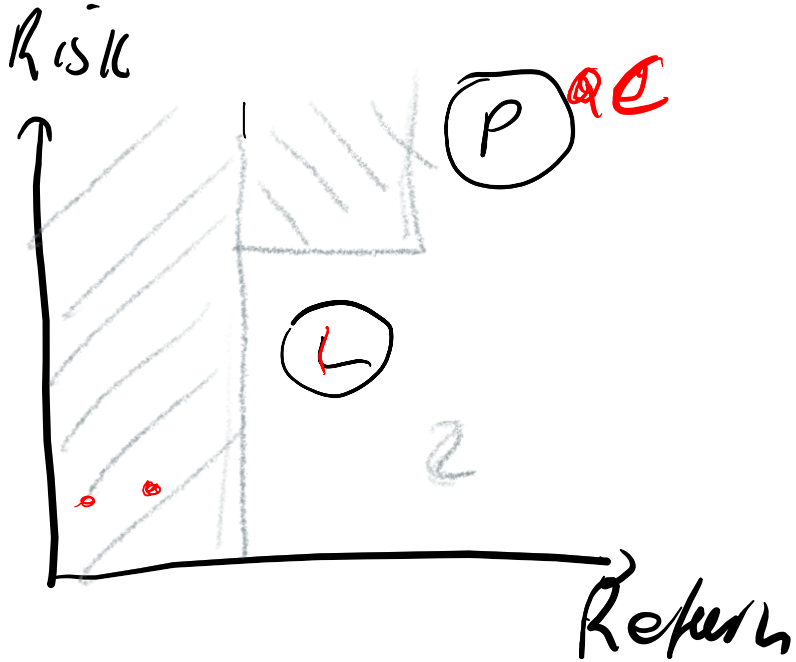
Mission

- Pioneering industrial research (P)
- Research on future products (L)
- Validation o

Excercise (via chat or audio)
What would you expect from an industrial research lab?

Porfolio Approach important

- Goal
- Life-cycle of research

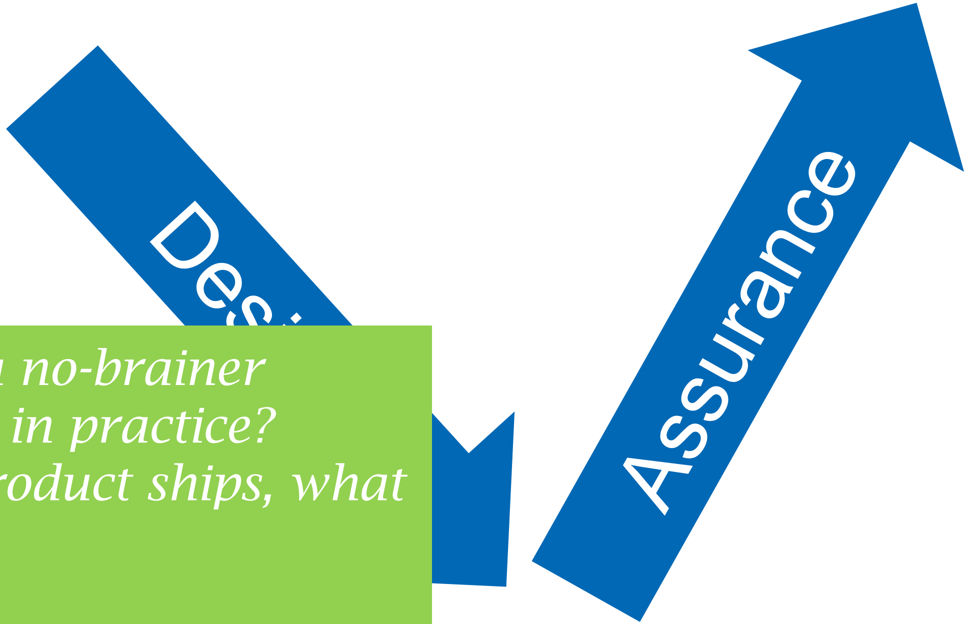


Outline

1. Security in practice – does anyone care?
2. Industrial vs. Basic Security Research
3. Example Research Projects
   a) Fuzzing Low-level Software
      Slides: Steffen Schulz, Brian Delgato
   b) Post Quantum Crypto
4. Discussion / Q&A

Secure Development Lifecycle (SDL)

- Some Goals:
  - Security by design
  - Well-defined security quality of all products
- V-Model
  - Design: Require
    Implementatio
  - **Assurance: Tes**
    Maintenance and CERT
  - Fuzzing as one tool to validate software



Excercise (chat): SDL seems a no-brainer
- *Why is it hard to establish in practice?*
- *Once a perfectly secure product ships, what can still go wrong?*

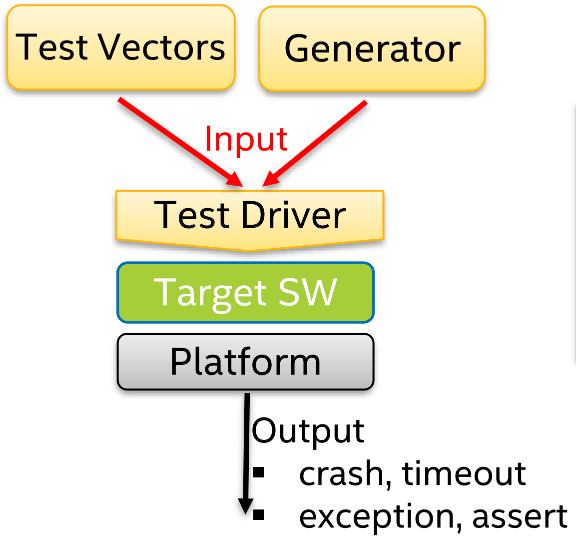
What is Fuzzing?

- “Smart“ randomized testing of software at scale
- Find inputs that crash, violate assertions or other checks

JPEG images generated from initial seed value „hello“
lcamtuf.blogspot.com/2014/11/pulling-jpegs-out-of-thin-air.html

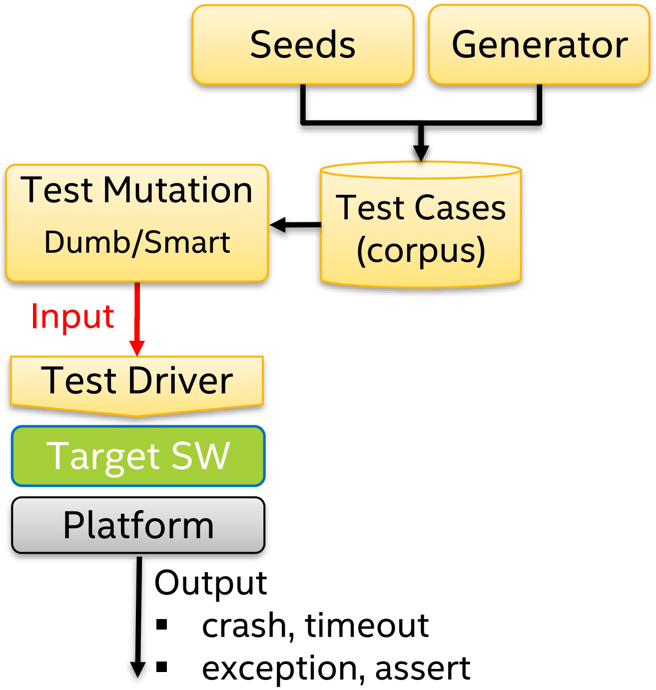
(Random) Testing

- Needs good inputs/generators
- High effort, small targets



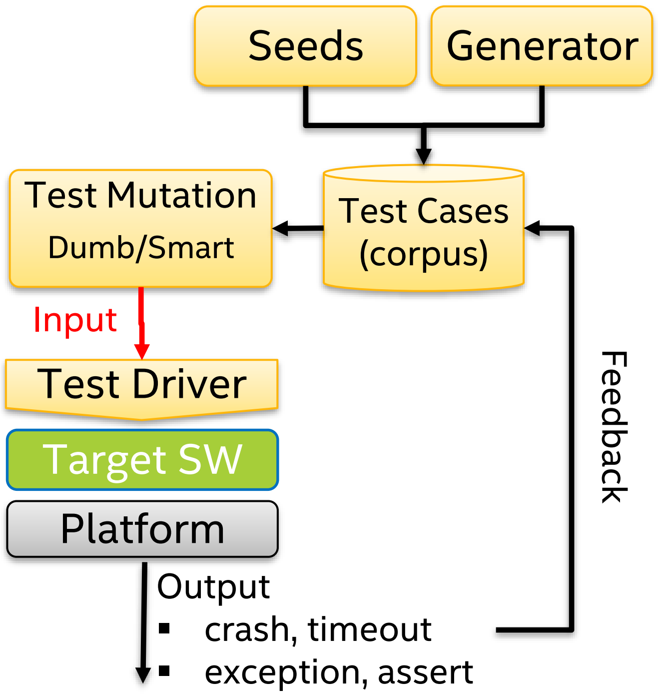
Mutation-based Fuzzing

- Expansion of simpler inputs/generators
- Execution focused around test corpus



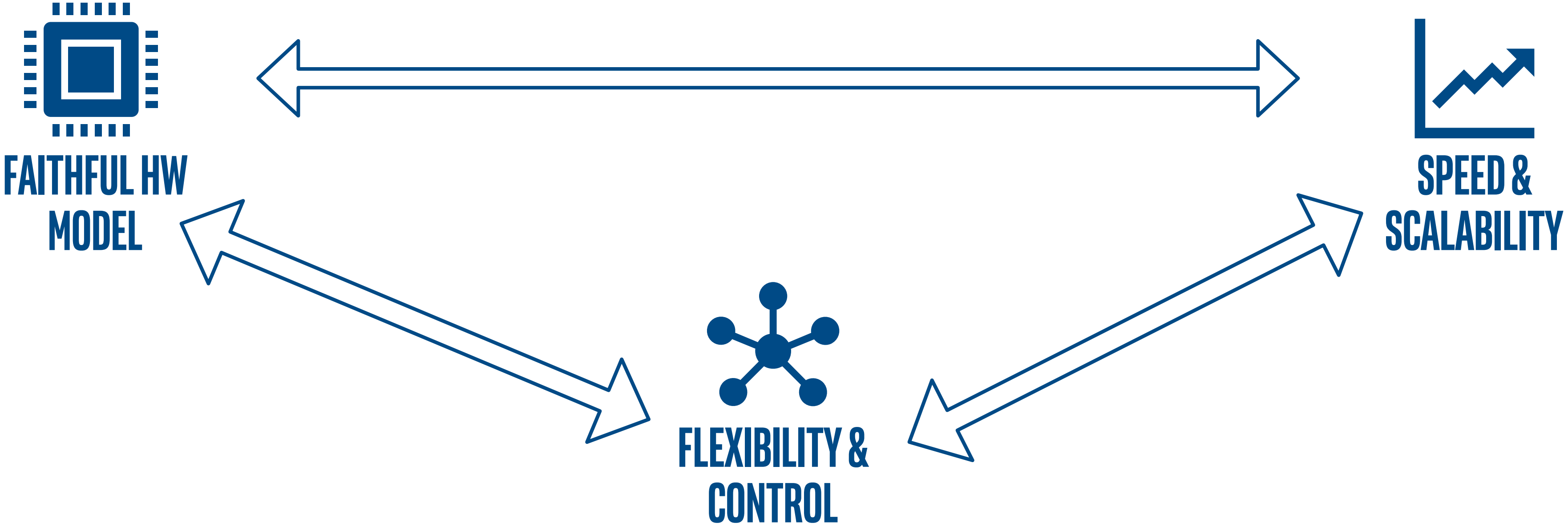
Feedback Guided Fuzzing

- Infinite, generic test expansion
- *Iteratively discover completely new inputs*



SUCCESS CRITERIA FOR SCALABLE FUZZING

- Although successful in software, fuzzing is difficult to deploy in low-level code
- It's also difficult to create an environment that is "real" and fast at the same time

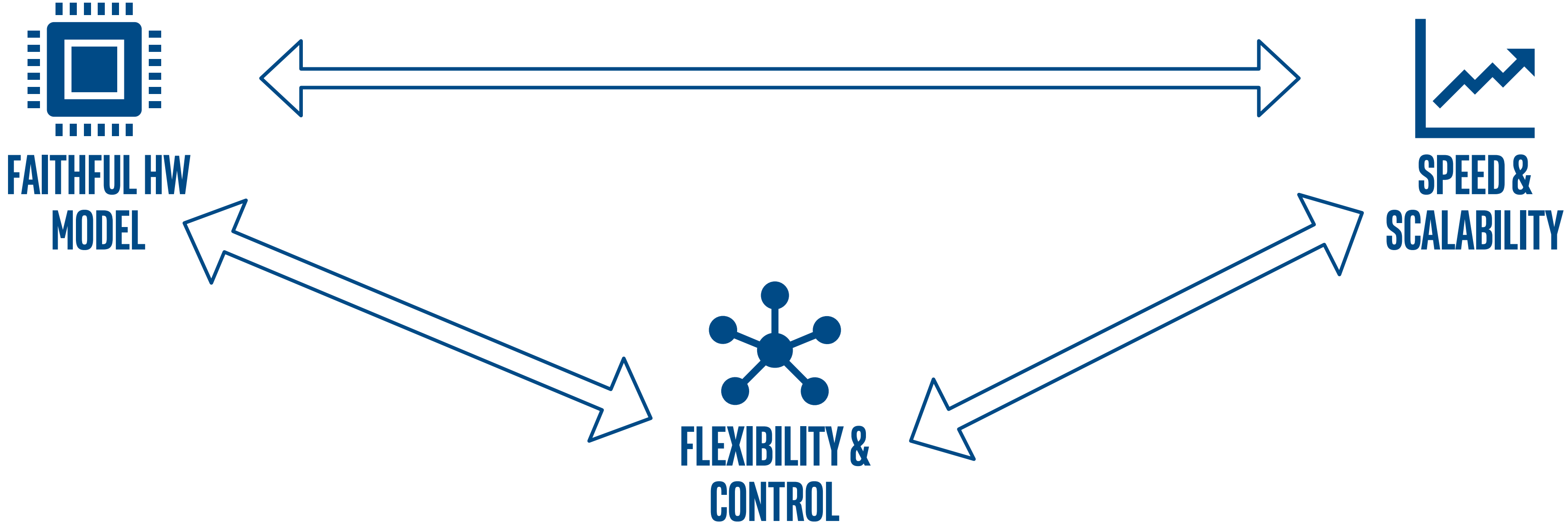


Citate as mentioned: Avatar, Pretender..

THREE APPROACHES FOR SCALABLE FUZZING

Excercise (chat): Advantages / Disadvantages of
- *Life platform = real hardware*
- *Emulation the hardware*
- *Re-hosting (running firmware as SW)*

- Although successful in software, fuzzing is difficult to deploy in low-level code
- It’s also difficult to create an environment that is “real” and fast at the same time

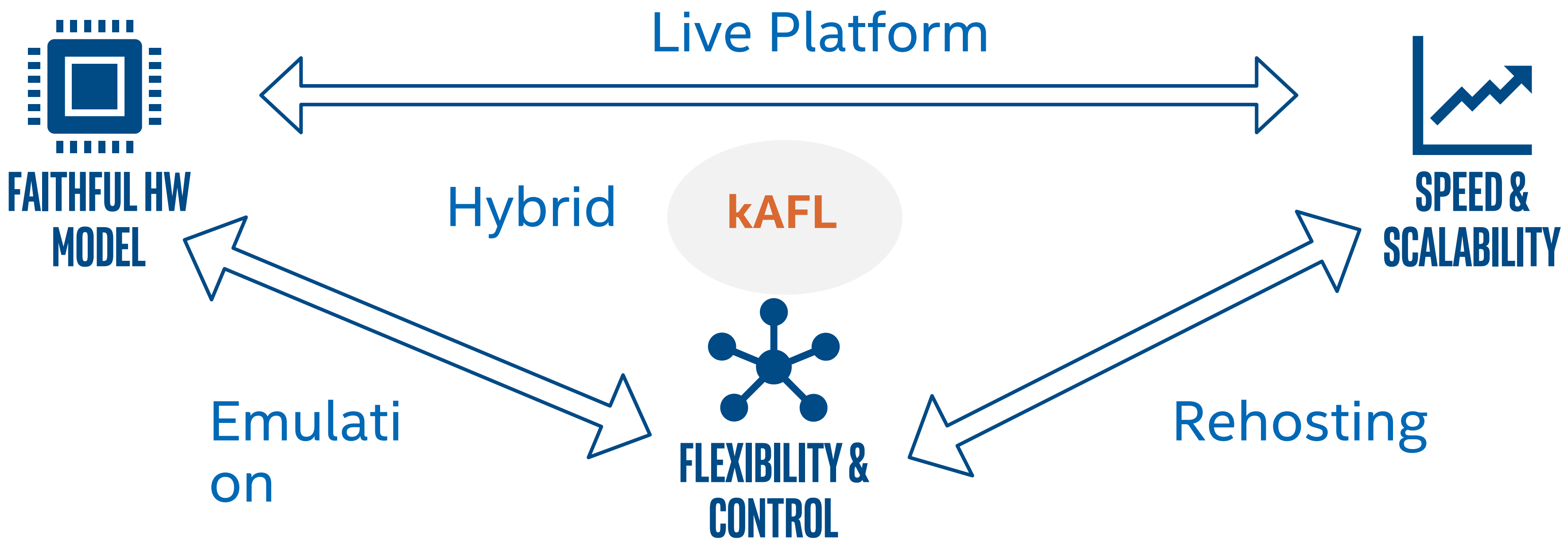


APPROACHES

1. Live HW
2. HW Emulation
3. Re-hosting

FIRMWARE FUZZING FOR VIRTUAL MACHINES

- Kernel AFL (kAFL): research vehicle developed at Ruhr-University Bochum
  - Accelerated execution & feedback using Intel® Virtualization Technology & Intel® Processor Trace features
  - Simple and fast, no assumptions on toolchain or target SW



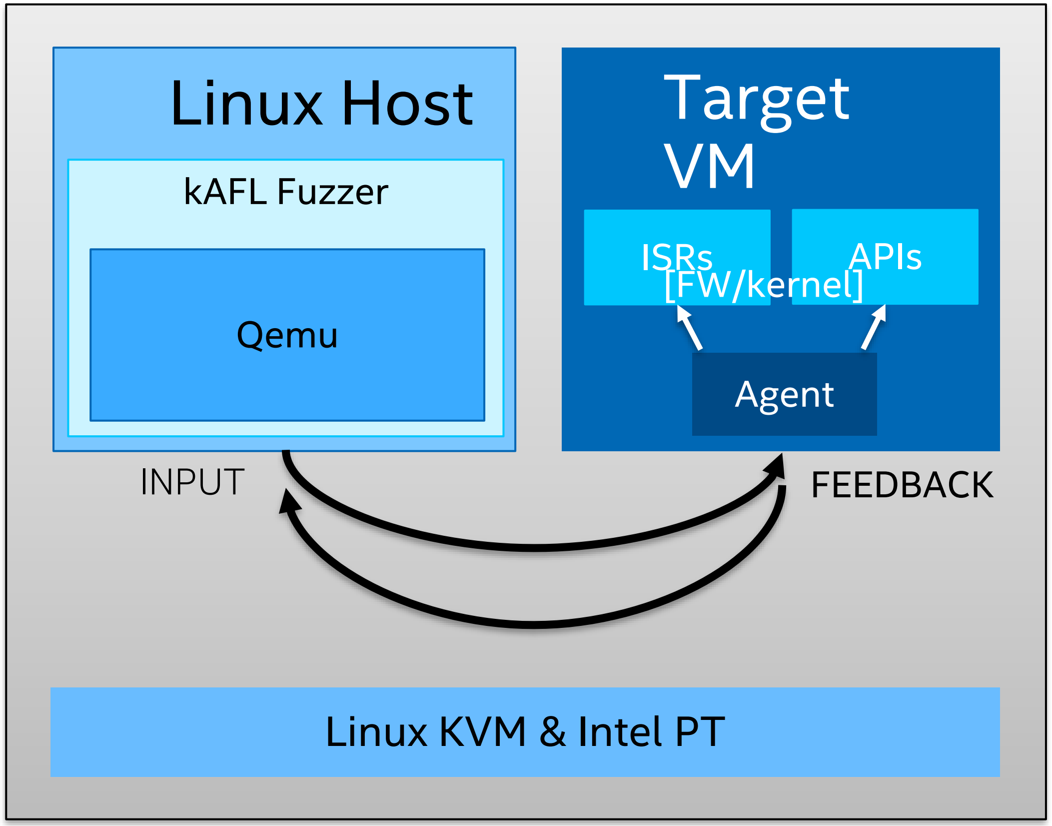
FIRMWARE FUZZING USING VIRTUAL MACHINES

Kernel AFL (kAFL) - No assumptions on toolchain or guest OS

- ✓ Usability: Easy to get started on UEFI, Zephyr RTOS and others
- ✓ Flexibility: Fast snapshot/reset, attach serial console or debugger
- ✓ Scalability: No special HW, work on laptop and scale on servers

X Open Problems

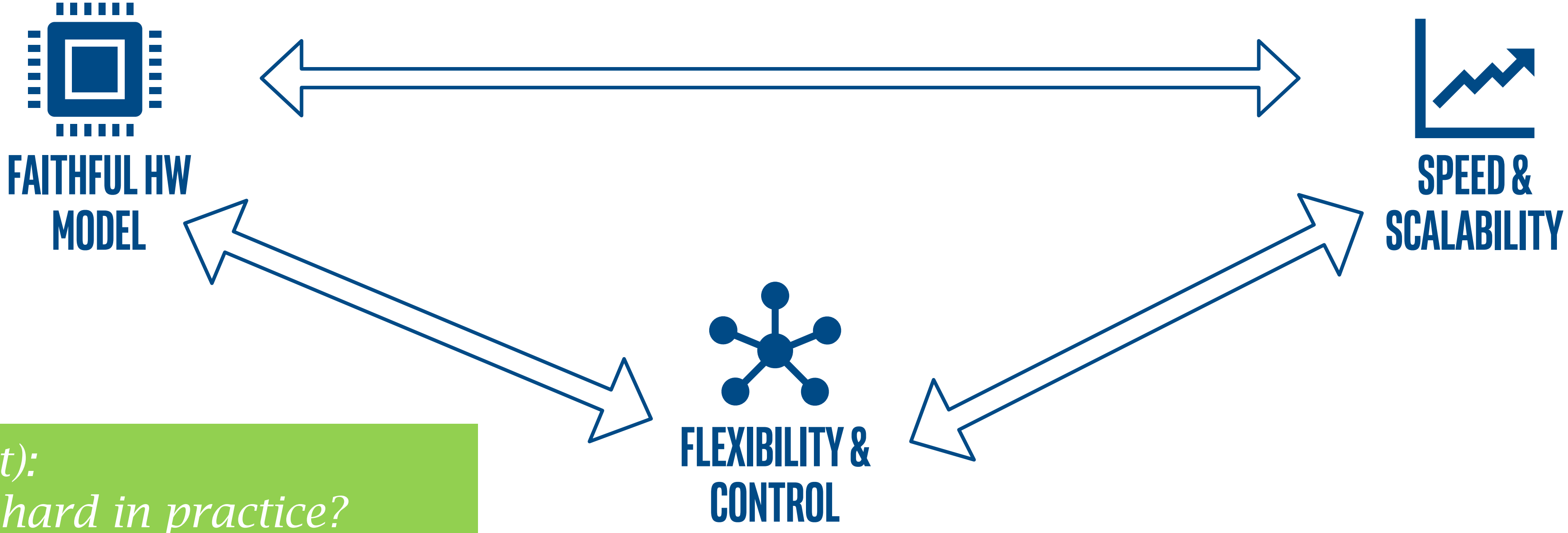
- How do we cope with unsupported devices/peripherals?



kAFL: Hardware-Assisted Feedback Fuzzing for OS Kernels – Schumilo et.al. Usenix 2017
https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/schumilo

RESEARCH CHALLENGE: I/O MODELING

- Faithful device models are a major bottleneck across all approaches
- Test focus typically on higher level parsing & processing, not I/O
- Can we overcome I/O dependencies for more scalable testing?
  - Generalize emulation, use machine learning, or other automation?



Excercise (chat):
- *Why is this hard in practice?*

PROCESS CHALLENGES

Continuous Integration environments can have thousands of check-ins a day

- How to fuzz effectively?
- Testing just the code being modified is helpful
- Tools like AFL Go (Bohme, M., Pham VT. et al) have potential to better target fuzzing

Complex tool-chains make it harder to change/modify compilers

How to handle triage at scale

- Fuzzers can provide a number of findings that require disposition
- How to better remove redundant findings?
- How much crash/hang analysis can be automated?

Outline

1. Security in practice – does anyone care?
2. Industrial vs. Basic Security Research
3. Example Research Projects
   a) Fuzzing Low-level Software
   b) Post Quantum Crypto
4. Discussion / Q&A

Break...

Post-Quantum World

Advances in the development of Quantum-Computers

Public-Key Cryptography is threatened

Post-Quantum Cryptography comes to the rescue

Quantum Attacks and Mitigations

- Symmetric Cryptography:
  - **Issue:** Grover's algorithm [Gro'96] is expected to break AES128 and SHA256
  - **Mitigation:** Increase keys/parameters of algorithms (Ex: AES128 → AES256)
- Public Key Crypto
  - **Issue**: Shor's alg and ECC
  - **Mitigation**: Repl encryption algorithms
- Replacements for

Excercise (chat): You are tasked to introduce PQC at Intel –

- *What obstacles do you expect?*
- *What hinders / accelerates adoption?*
- *Where would you start?*

Quantum Cryptography:

- ✓ Uses quantum physics to achieve higher security
- ✗ Requires quantum infrastructure
- ✗ Restricted to Key Exchange (e.g., [BB84])
- ✗ No standards

Post-Quantum Cryptography:

- ✓ Based on harder math problems
- ✓ Can be implemented in current infrastructure
- ✓ Offers all required features
- ⚠ Standards under development

Changing Tires on a Moving Car

- PQC transition is an unprecedented move – Crypto adoption takes decades
  - Standards are being defined at the same time cryptanalysis is being understood
  - Post-Quantum Crypto literature may not offer drop-in replacements for all features

Post-Quantum Cryptography Families

More Mature

Less Mature

Hash Based Signatures

Security: relies on well-known security notions

Code Based Cryptography

Digital Signatures

Security: (presumably well-known) problems from coding-theory

Lattices Based Cryptography

Security: (presumably well-known) problems from lattices

Multivariate Cryptography

Encryption, Key Exchange, Signatures

Security: other problems from multivariate quadratic equations

Isogeny Based Cryptography

Encryption, Key Exchange, Signatures

Security: other problems from isogenies of super-singular elliptic curves

Digital Signatures

Key Exchange, Signatures

6

Remarks

- PQC transistion is an unprecedented move
- Industry perspective is critical for wide adoption
  - Ease of deployment
  - Scalability
  - Maintenance
- Simple & well-understood is better than complex & less-understood
- Standards are much needed but we should not rush at the cost of security

Outline

1. Security in practice – does anyone care?
2. Industrial vs. Basic Security Research
3. Example Research Projects
   a) Fuzzing Low-level Software
   b) Post Quantum Crypto
4. Discussion / Q&A

This is the last slide…
- *Any questions?*
- *Any feedback/comments (chat/audio)?*