Debunking Fault Injection Myths and Misconceptions

Niek Timmers
niek@twentytwosecurity.com
@tieknimmers

Cristofaro Mune
c.mune@pulse-sec.com
@pulsoid

Today’s agenda

- Introduction
- Fault injection, what is it?
- Fault injection, where are we now?
- Trends
- Debunking myths
- Takeaways

Who are we…

- Cristofaro Mune
  - Product Security Consultant
  - Security trainer
  - Research:
    - Fault injection
    - TEEs
    - White-box Cryptography
    - Device exploitation
- Niek Timmers
  - Freelance Device Security Expert
  - Security trainer
  - Interests:
    - Embedded device security
    - Secure boot
    - Hardware attacks
    - Automotive

WHAT IS FAULT INJECTION?

Fault injection basics

"Introducing faults into a chip to alter its intended behavior."

```
// check if secure boot is enabled
if(OTP_READ(SECURE_BOOT) & 1) {
    authenticate(image)
}

// start image
execute_image(&image);
```

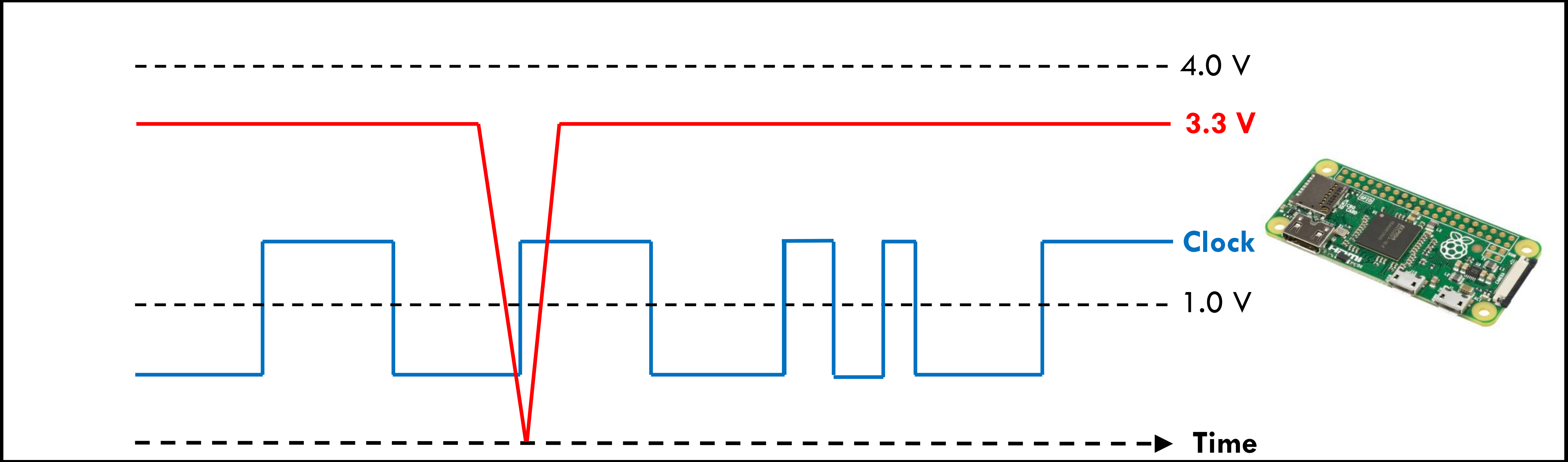
How do you introduce those faults?

Fault injection techniques

Faults are introduced by injecting glitches that put a chip temporarily outside of its expected conditions.

Fault injection techniques

Faults are introduced by injecting glitches that put a chip temporarily outside of its expected conditions.

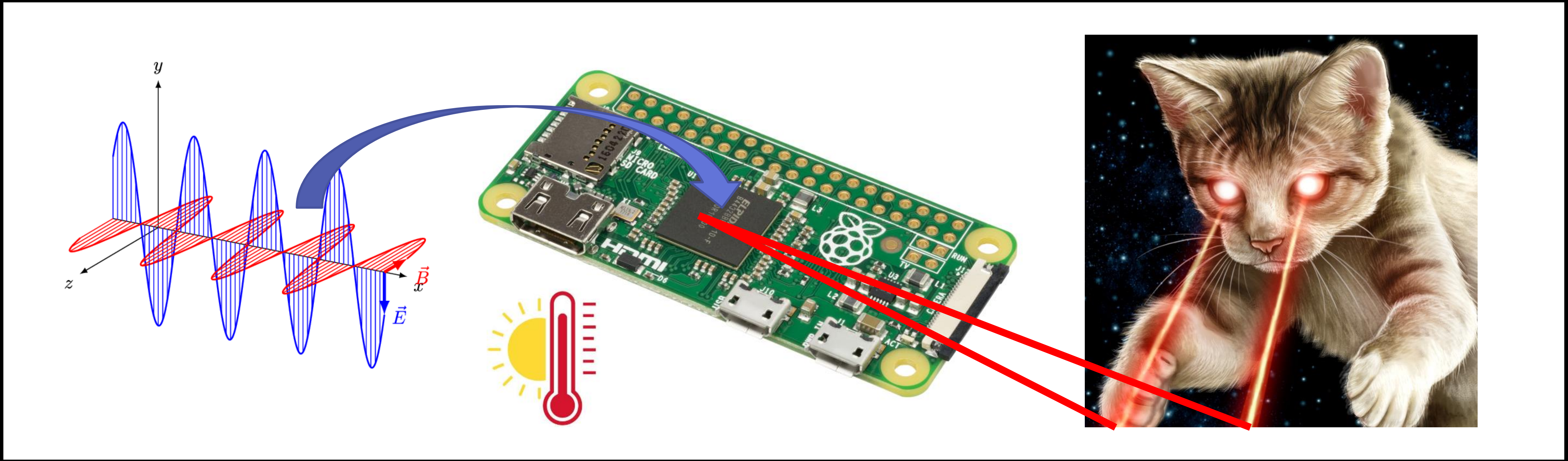


Voltage

Clock

Fault injection techniques

Faults are introduced by injecting glitches that put a chip temporarily outside of its expected conditions.



Voltage Clock Electromagnetic Laser

WHERE ARE WE NOW?

Research

- There's academic conferences

FDTC 2018
Fault Diagnosis and Tolerance in Cryptography

- Great academic papers at various conferences

The Sorcerer's Apprentice Guide to Fault Attacks

Hagai Bar-El[1] Hamid Choukri[2,3] David Naccache[3] Michael Tunstall[3,4] Claire Whelan[5]

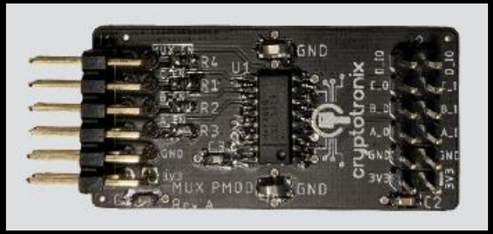
- Great contributions from the community at various conferences
  - E.g. Exide @ REcon 2014

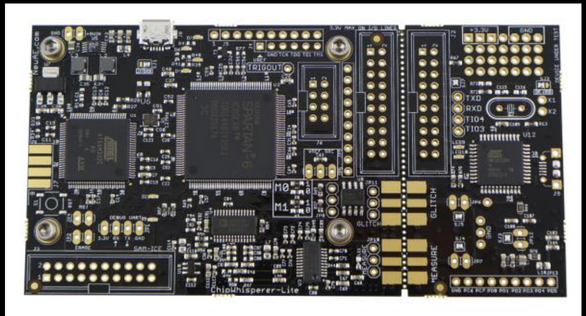
Glitching For n00bs

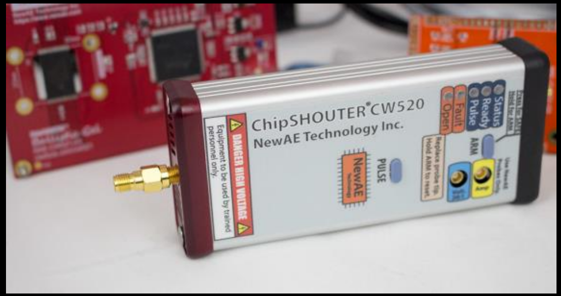
A Journey to Coax Out Chips' Inner Secrets

Tooling

- Do-it-yourself
  - < $100 (Voltage)
  - E.g. chipfail glitcher
- Commercial (affordable)
  - < $1000 (Voltage); < $4000 (EMFI)
  - E.g. NewAE ChipWhisperer
- Commercial (professional)
  - > $10,000 (Voltage, EMFI, Laser, etc.)
  - E.g. Riscure Inspector FI



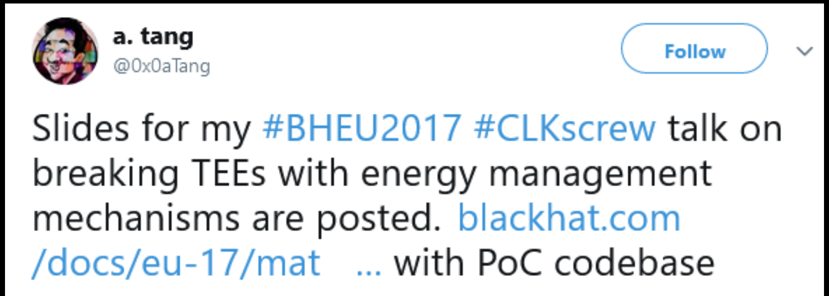




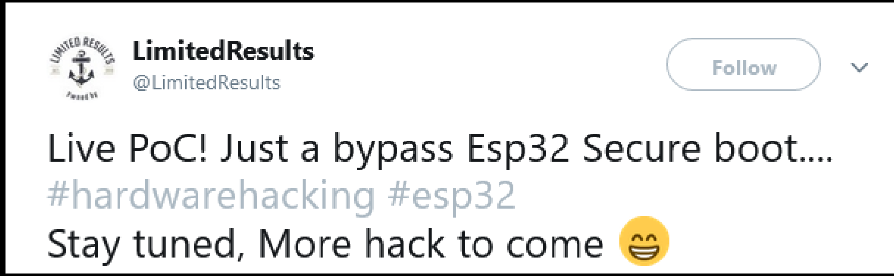
Attacks

- Breaking the security of crypto wallets

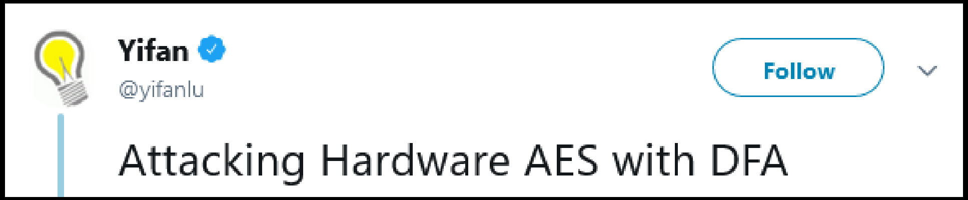
- Breaking the security of smart phones



- Breaking the security of secure boot



- Breaking the security of crypto engines

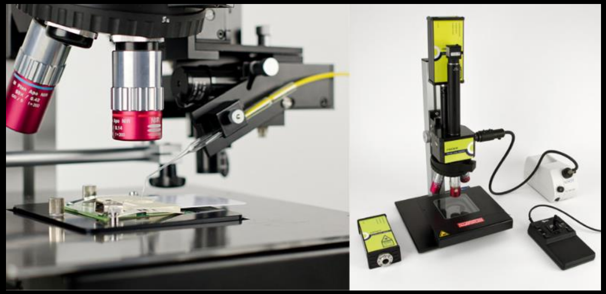


Trends

- Tooling is becoming available to the masses
- Lots of focus on the ‘how to inject a glitch’ part of an attack
- Most research conducted on low power chips
- Focus is mostly on altering software behavior

Important exceptions

- Optical fault injection tooling not available to the masses



- Academia performs theoretical research on fault injection

Electromagnetic fault injection: towards a fault model on a 32-bit microcontroller

Nicolas Moro*‡, Amine Dehbaoui†, Karine Heydemann‡, Bruno Robisson*, Emmanuelle Encrenaz‡
**Commissariat à l'Énergie Atomique et aux Énergies Alternatives (CEA)*
Gardanne, France

- Real attackers go further than:
  - low powered chips
  - just altering software

So, we glitch the Switch and get the keys...?

@qlutoo, @derrekr6, @naehrwert

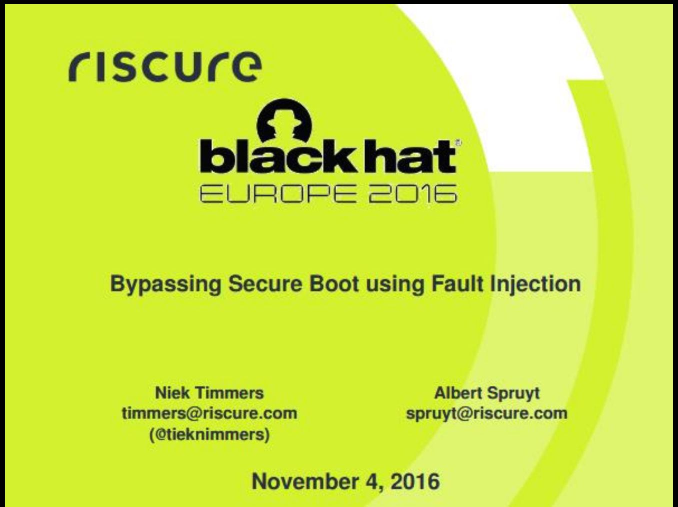
WHERE DO WE FIT IN?

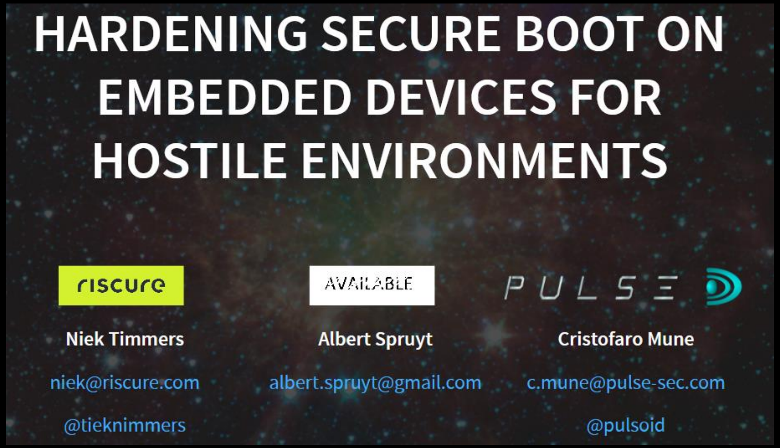
What we are working on…

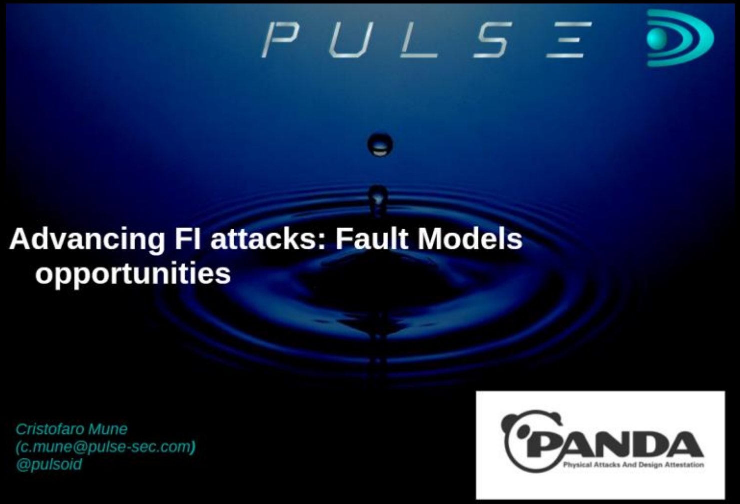
- A fault injection think tank (AllOurFaults):
  - Alyssa Milburn (@noopwafel)
  - Albert Spruyt
  - Cristofaro Mune (@pulsoid)
  - Niek Timmers (@tieknimmers)
- An open source voltage glitching platform
- Fault injection research; some results covered in this presentation
- You can find us on: allourfaults.com and @allourfaults

Published fault injection research

- Academic contributions:
  - [Controlling PC on ARM using Fault Injection, 2016]
  - [Escalating Privileges in Linux using Voltage Fault Injection, 2017]
- Several community contributions:





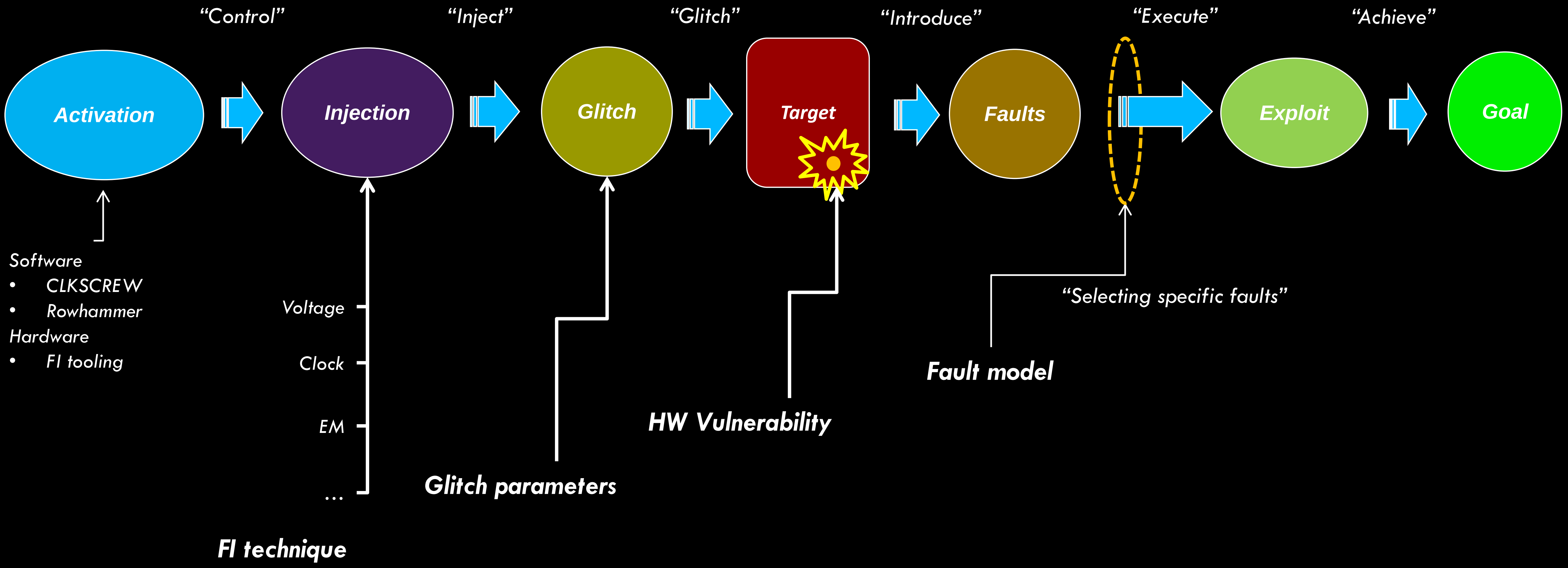


Lots of research…

but still many ‘Myths and Misconceptions’

Let’s debunk them in a systematic fashion!

Fault injection reference model



Here they come…

"Fault attacks are not effective on >1 GHz chips."

Escalating Privileges in Linux using Voltage Fault Injection

Niek Timmers
Riscure – Security Lab
timmers@riscure.com / @tieknimmers

Cristofaro Mune
Embedded Security Consultant
pulsoid@icysilence.org / @pulsoid

All attacks are demonstrated using a commercially available development board, from now on referred to as *Target*, which is designed around a fast and feature rich ARM Cortex-A9 processor SoC. A commercially available V-FI test bench is used to perform *V-FI* on the *Target*. The processor operates at 1 GHz and the *instruction cache* and *data cache* are by default enabled. All attacks described in this paper are executed from external DDR3 unless cached.

FAULT ATTACKS ARE NOT EFFECTIVE ON > 1 GHZ CHIPS

DEBUNKED

BUT THAT’S VOLTAGE… WHAT ABOUT EMFI?

“EMFI does not work on >100 MHz chips.”

- Awesome do-it-yourself EMFI tool
- Incorrect statement on EMFI attacks
- Not everybody aware of EMFI research

WOOT '17

11th USENIX Workshop on Offensive Technologies

AUGUST 14-15, 2017
VANCOUVER, BC, CANADA
Co-located with USENIX Security '17

in modifiying the control flow of processors. Moro et al. [10] were able to successfully modify the control flow of an ARM Cortex-M3 processor through both instruction modification and stepping. However, despite advances in EMFI technology, thus far EMFI attacks against modern gigahertz-speed are absent in literature. A survey of attacks and countermeasures suggests that 100 MHz is the state of the art in the field of EMFI attacks.

“BADFET: Defeating Modern Secure Boot Using Second-Order Pulsed Electromagnetic Fault Injection” – *Cui, Housley*

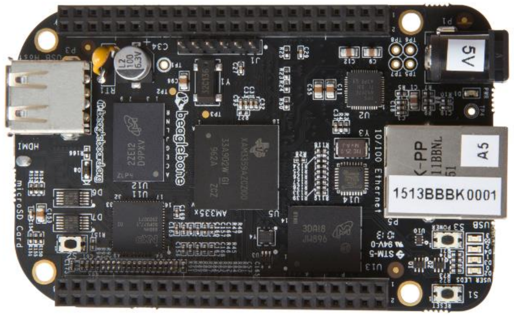
Actually…

Exploring Effects of Electromagnetic Fault Injection on a 32-bit High Speed Embedded Device Microprocessor

Tim Hummel

July 27, 2014

to be an ARM and it has to implements trace functionality. We selected the Beagle Bone Black (BBB) development board. The BBB has an AM3358 family processor, the Texas Instruments Sitara AM3358AZCZ100 [Ins] microprocessor. It contains an A8 running with up to 1 Ghz clock speed. This 1 Ghz maximum clock speed was used in all our experiments. Figure 4.1 shows a top view of the board, the processor is in the square package "U5" in the middle of the board. This target fulfills all necessary requirements needed for glitchability and glitch effect analysis.



ElectroMagnetic Fault Injection Characterization

George Thessalonikefs
george.thessalonikefs@os3.nl

University of Amsterdam
System & Network Engineering MSc

February 10, 2014

2.2 Target

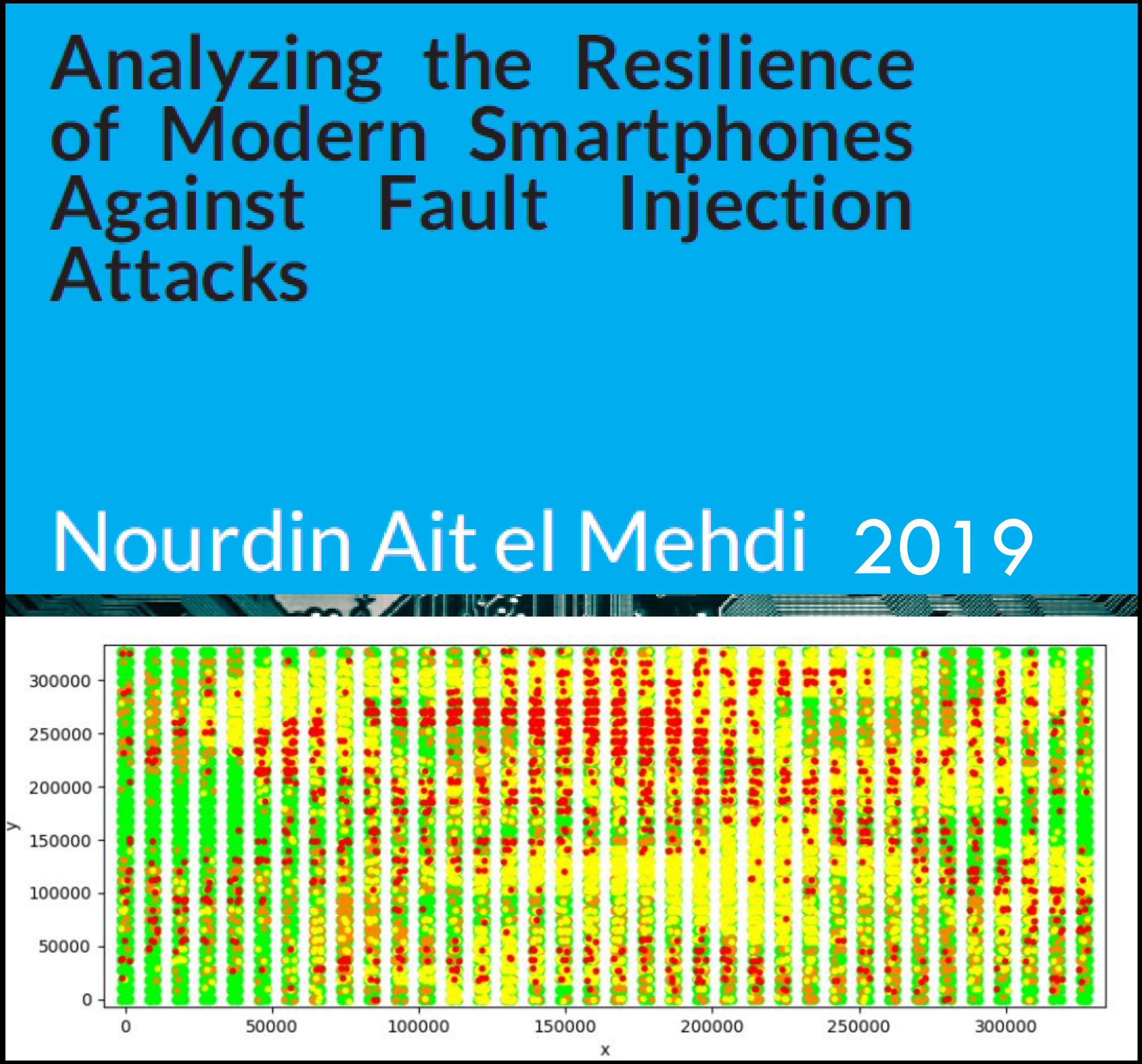
The target of the research is the 32-bit ARM Cortex-A9 processor which implements the ARMv7-A architecture based on the RISC architecture. The Cortex-A9, being one of the state of the art processors used in smartphones, tablets, home media players, etc, has many advanced features (such as floating point processing engine) that will not be used during this research. Thus, features of the Cortex-A9 processor relative to the research include:

Clock speed
The Cortex-A9 was used with a clock speed of 792 MHz. This results in approximately 1,26 nanoseconds per clock cycle.

Glitches were found to take place in the fetch, decode, execute and write-back phases of the pipeline. The results of those glitches were instruction skipping, MMU exceptions followed by a reset issued by the processor, and wrong value on the output register. The latter presented a tendency to transition bits from '1' to '0'.

Attacks above 100 MHz already published in 2014…

More EMFI research above 100MHz



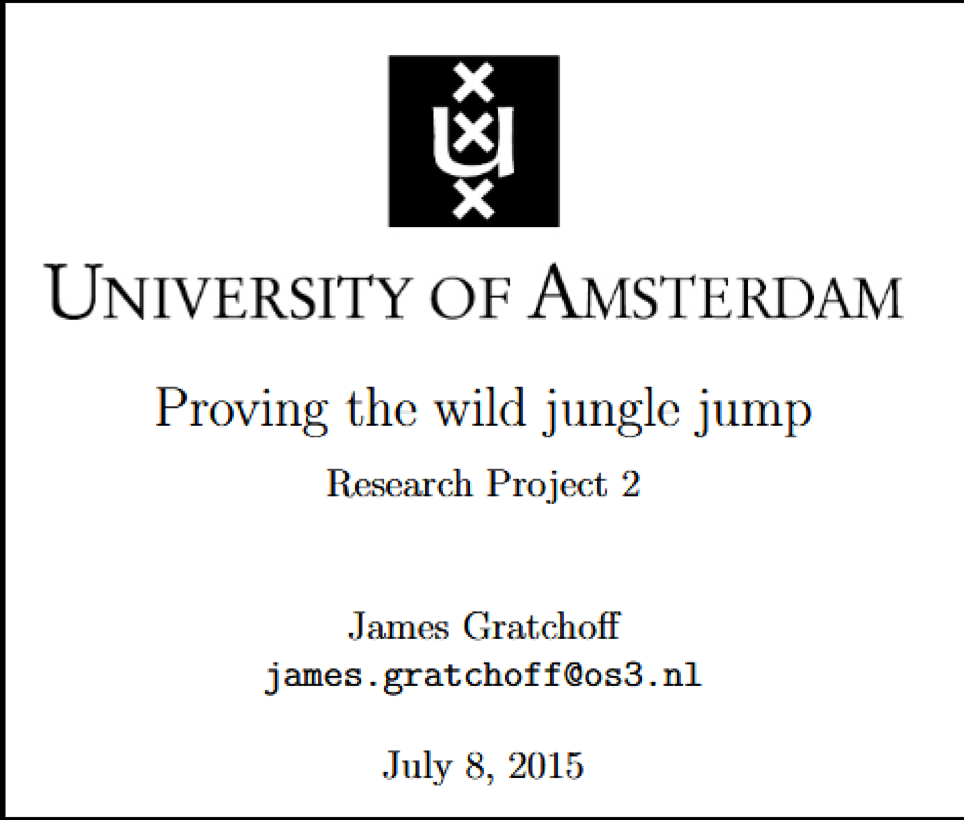
EM-FI DOES NOT WORK ON >100 MHZ TARGETS

Research Fragmentation

- Fault injection research is conducted in multiple communities:
  - Academia
  - Industry
  - Security community
- Consolidation of knowledge does not always happens
- Result: Research is being missed

Inconsistent views result in ‘Myths and Misconceptions’

“Fault attacks are used to bypass SW checks”

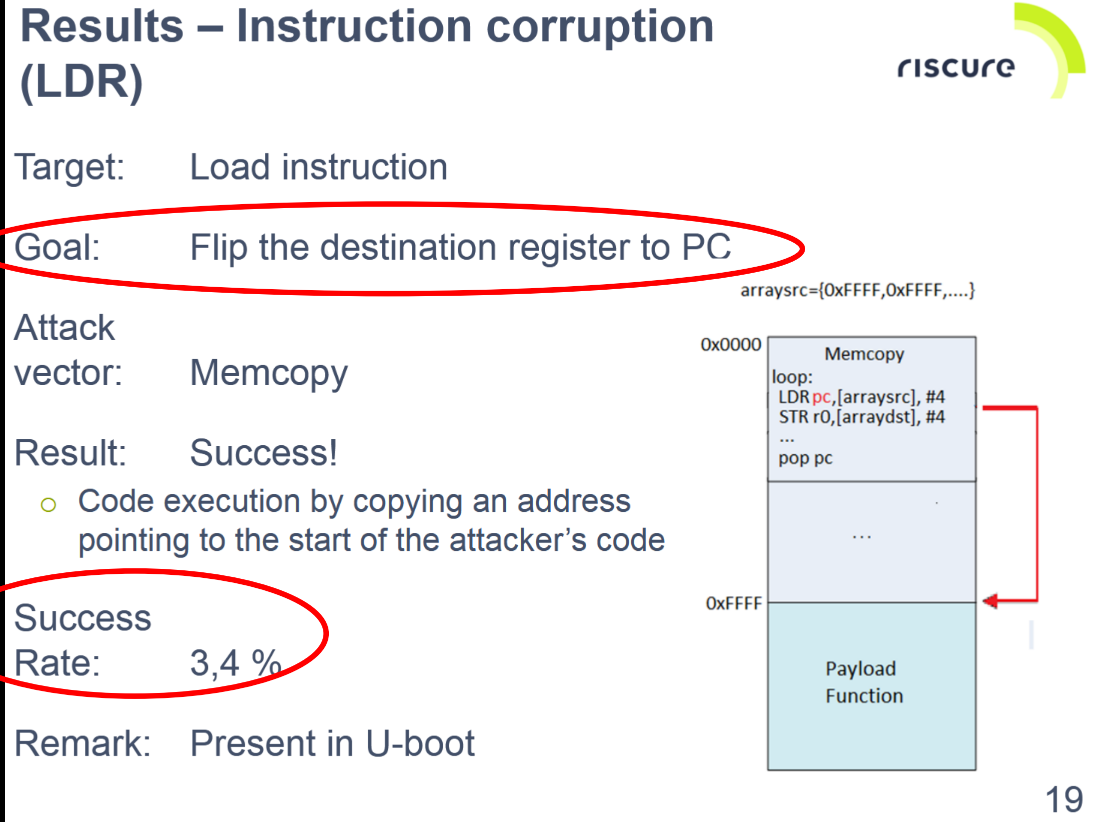

UNIVERSITY OF AMSTERDAM

Proving the wild jungle jump

Research Project 2

James Gratchoff
james.gratchoff@os3.nl

July 8, 2015



Report / Slides

"Fault attacks are used to bypass SW checks"

Preset user space registers.

```
. . .
int rand = random();
*(volatile unsigned int *)(trigger) = HIGH;

volatile (
  "movw r12, #0x4141;" // Repeat for other
  "movt r12, #0x4141;" // unused registers
  . . .
  "mov r7, %[rand];" // Random syscall nr
  "swi #0;"          // Linux kernel takes over
  . . .

*(volatile unsigned int *)(trigger) = LOW;
. . .
```

Linux Kernel Privilege Escalation

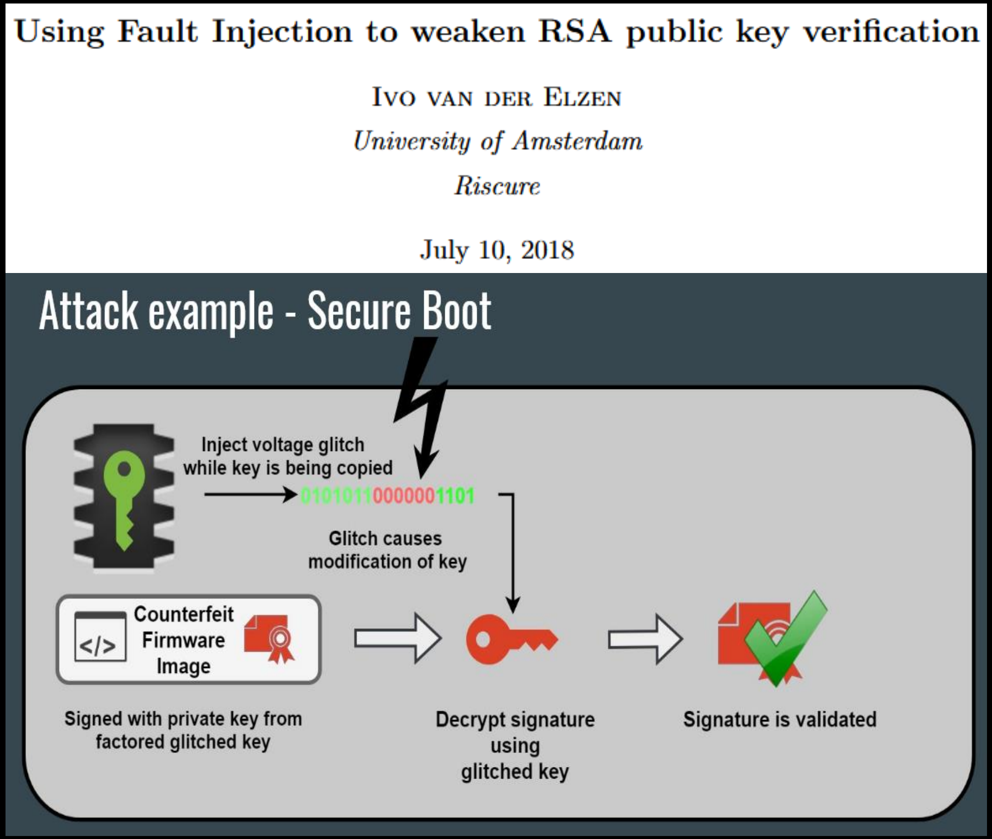
```
Unable to handle kernel paging request at virtual addr 41414140
pgd = 5db7c000..[41414140] *pgd=0141141e(bad)
Internal error: Oops - BUG: 8000000d [#1] PREEMPT SMP ARM
Modules linked in:
CPU: 0 PID: 1280 Comm: control-pc Not tainted <redacted> #1
task: 5d9089c0 ti: 5daa0000 task.ti: 5daa0000
PC is at 0x41414140
LR is at SyS_prctl+0x38/0x404
pc : 41414140  lr : 4002ef14  psr: 60000033
sp : 5daa1fe0  ip : 18c5387d  fp : 41414141
r10: 41414141  r9 : 41414141  r8 : 41414141
r7 : 000000ac  r6 : 41414141  r5 : 41414141  r4 : 41414141
r3 : 41414141  r2 : 5d9089c0  r1 : 5daa1fa0  r0 : ffffffea
```

Control of kernel PC from user space!

"Don't tell anyone…No checks involved!"

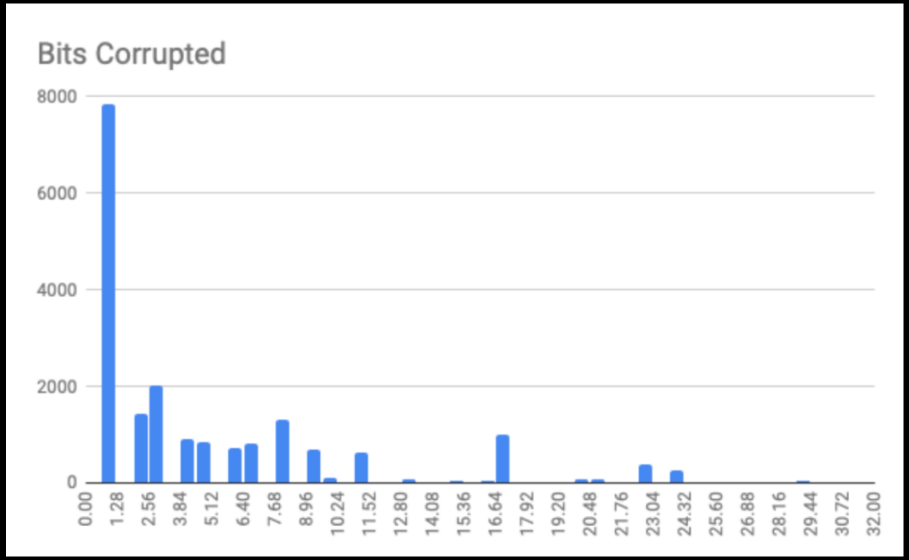
“Fault attacks are used to bypass SW checks”

- RSA key weakening by flipping bits in the modulus
- Also performed as part of other attacks:
  - E.g CLKSCREW



“Fault attacks are used to bypass SW checks”

- PlayStation Vita attack
  - Differential Fault Analysis Attack (DFA) on cryptographic engines
- Recovered keys from the target
  - 30 master keys
  - 238 out of 240 non-master keys



Yifan Lu – “Attacking Hardware AES with DFA” – (PS Vita)
Paper/Blog

FAULT ATTACKS ARE USED TO BYPASS SW CHECKS

“Fault attacks are not effective on multi-core chips.”

- Multiple cores have an impact…but fault injection still possible.
- Even when cores verify each other in lockstep

Safety ≠ Security

A security assessment of the resilience against fault injection attacks in ASIL-D certified microcontrollers

Nils Wiersma, Ramiro Pareja
Riscure Security Lab
{wiersma, pareja} @ riscure.com

Of all the safety mechanisms implemented in ASIL-D MCUs, we are only interested in investigating the ones that have an effect on transient faults as they could also mitigate the glitches used by an FI attacker. In both selected targets these mechanisms include a dual core CPU in lockstep configuration (or ‘Simple Time Redundancy with Comparison’) and memories with error correction codes (ECC) and parity bits, as recommended by ISO 26262 part 5.

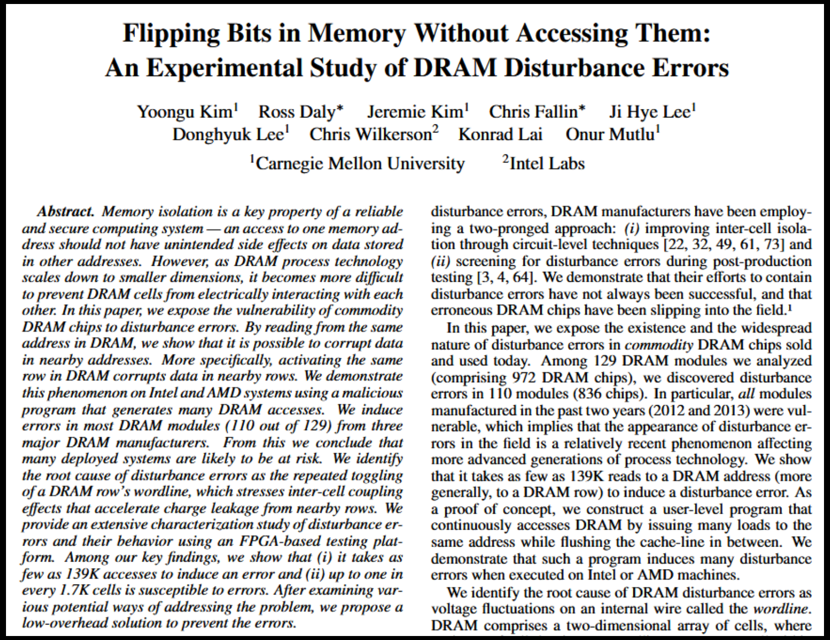
TABLE II
THE SUCCESS RATES OF THE CHARACTERIZATION EXPERIMENTS

	unroll		auth	
	Power	EM	Power	EM
ASILD1	87%	0.2%	60%	0.2%
ASILD2	0%	18%	N/A	57%
QM1	100%	N/A	N/A	N/A

FAULT ATTACKS DO NOT WORK ON MULTI-CORE CHIPS

“Physical access is required to perform fault attacks.”

Use case #1: Rowhammer



Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors

Yoongu Kim[1] Ross Daly* Jeremie Kim[1] Chris Fallin* Ji Hye Lee[1]
Donghyuk Lee[1] Chris Wilkerson[2] Konrad Lai Onur Mutlu[1]
[1]Carnegie Mellon University [2]Intel Labs

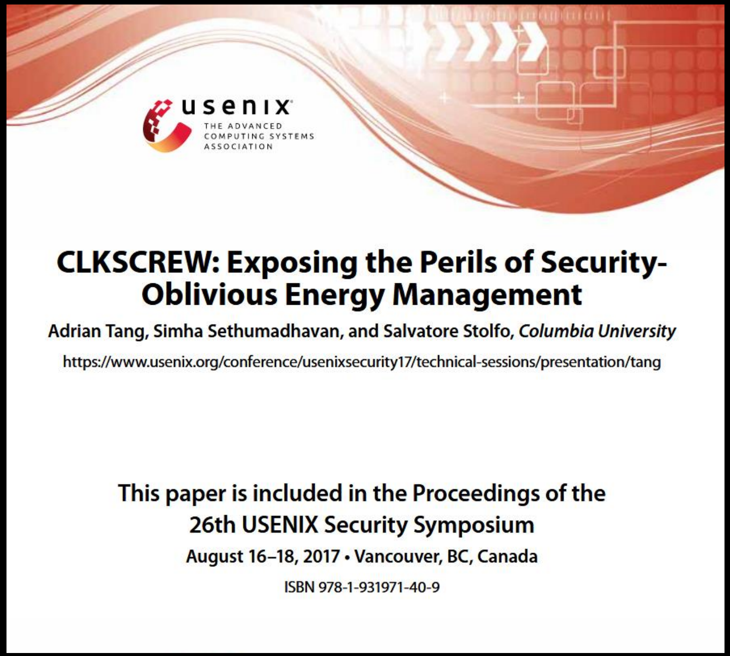
Abstract. *Memory isolation is a key property of a reliable and secure computing system — an access to one memory address should not have unintended side effects on data stored in other addresses. However, as DRAM process technology scales down to smaller dimensions, it becomes more difficult to prevent DRAM cells from electrically interacting with each other. In this paper, we expose the vulnerability of commodity DRAM chips to disturbance errors. By reading from the same address in DRAM, we show that it is possible to corrupt data in nearby addresses. More specifically, activating the same row in DRAM corrupts data in nearby rows. We demonstrate this phenomenon on Intel and AMD systems using a malicious program that generates many DRAM accesses. We induce errors in most DRAM modules (110 out of 129) from three major DRAM manufacturers. From this we conclude that many deployed systems are likely to be at risk. We identify the root cause of disturbance errors as the repeated toggling of a DRAM row's wordline, which stresses inter-cell coupling effects that accelerate charge leakage from nearby rows. We provide an extensive characterization study of disturbance errors and their behavior using an FPGA-based testing platform. Among our key findings, we show that (i) it takes as few as 139K accesses to induce an error and (ii) up to one in every 1.7K cells is susceptible to errors. After examining various potential ways of addressing the problem, we propose a low-overhead solution to prevent the errors.*

disturbance errors, DRAM manufacturers have been employing a two-pronged approach: *(i)* improving inter-cell isolation through circuit-level techniques [22, 32, 49, 61, 73] and *(ii)* screening for disturbance errors during post-production testing [3, 4, 64]. We demonstrate that their efforts to contain disturbance errors have not always been successful, and that erroneous DRAM chips have been slipping into the field.[1]

In this paper, we expose the existence and the widespread nature of disturbance errors in *commodity* DRAM chips sold and used today. Among 129 DRAM modules we analyzed (comprising 972 DRAM chips), we discovered disturbance errors in 110 modules (836 chips). In particular, *all* modules manufactured in the past two years (2012 and 2013) were vulnerable, which implies that the appearance of disturbance errors in the field is a relatively recent phenomenon affecting more advanced generations of process technology. We show that it takes as few as 139K reads to a DRAM address (more generally, to a DRAM row) to induce a disturbance error. As a proof of concept, we construct a user-level program that continuously accesses DRAM by issuing many loads to the same address while flushing the cache-line in between. We demonstrate that such a program induces many disturbance errors when executed on Intel or AMD machines.

We identify the root cause of DRAM disturbance errors as voltage fluctuations on an internal wire called the *wordline*. DRAM comprises a two-dimensional array of cells, where

Use case #2: CLKSCREW



CLKSCREW: Exposing the Perils of Security-Oblivious Energy Management

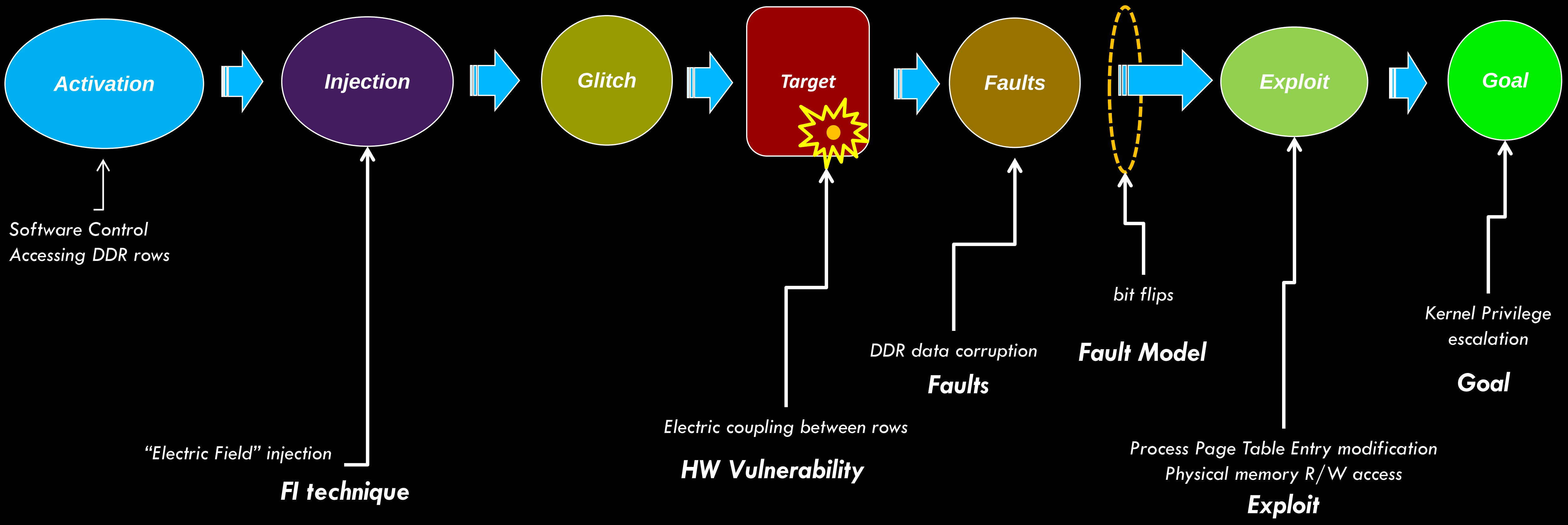
Adrian Tang, Simha Sethumadhavan, and Salvatore Stolfo, *Columbia University*

https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/tang

This paper is included in the Proceedings of the
26th USENIX Security Symposium
August 16–18, 2017 • Vancouver, BC, Canada
ISBN 978-1-931971-40-9

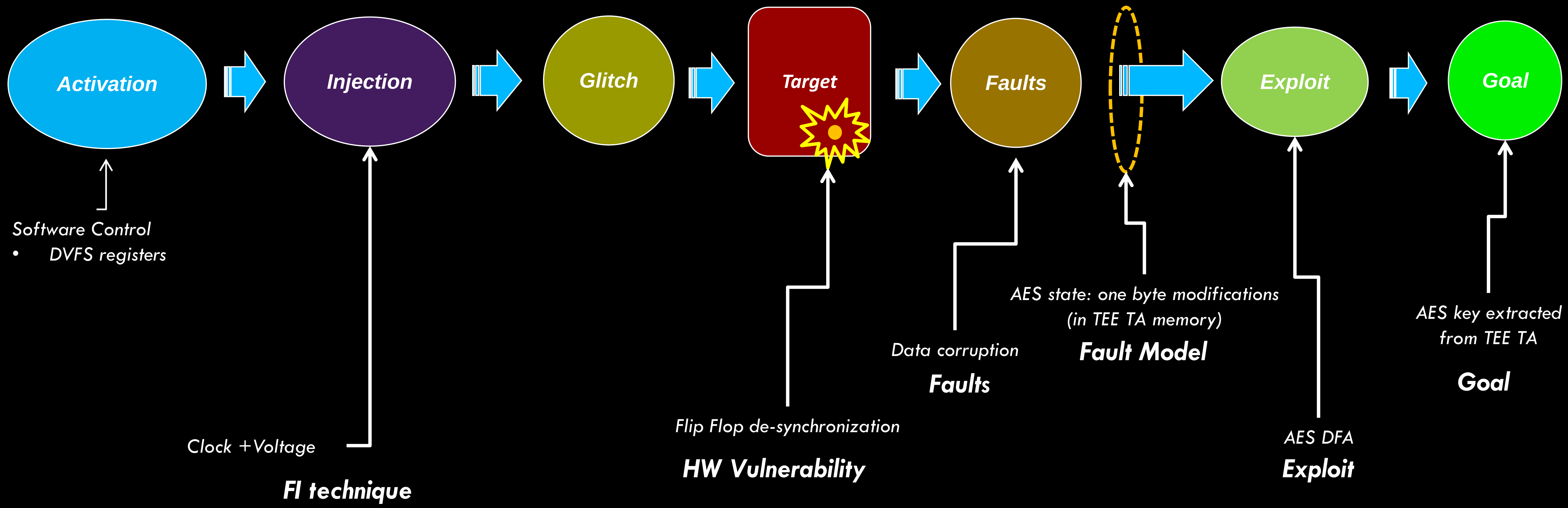
These HW vulnerabilities can be remotely triggered by software

Rowhammer: Kernel Privilege Escalation



Reference: Google Project Zero

CLKSCREW: Key extraction



Reference: Clkscrew paper

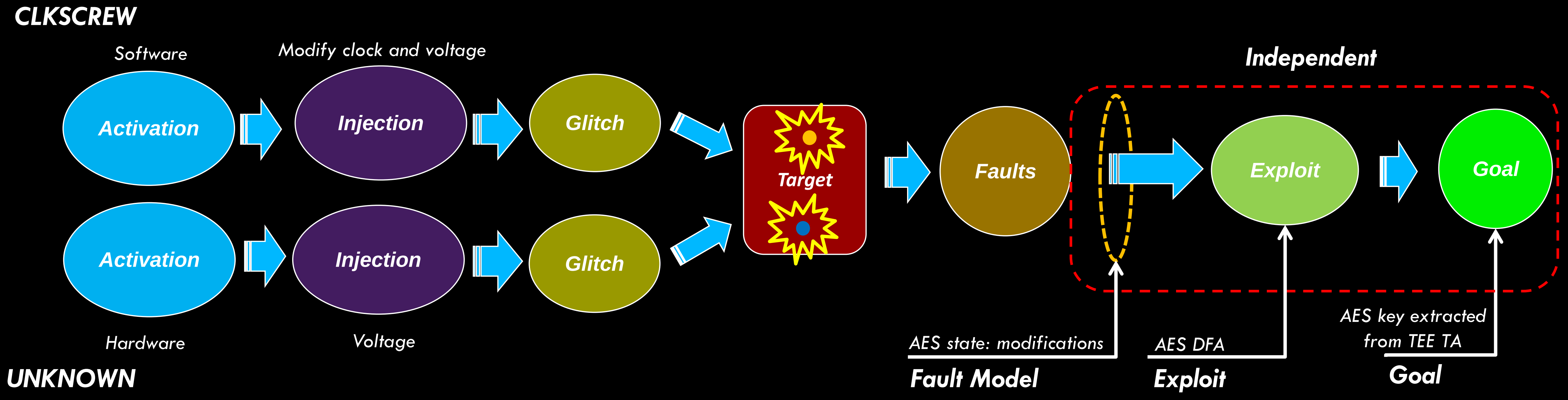
PHYSICAL ACCESS REQUIRED FOR FI

DEBUNKED

“Fault attacks are injection dependent.”

- Literature often links injection technique to goal:
  - E.g. “Fault injection technique A is used for attack B”
- No systematic comparison of faults available
- Actually… specific fault models are applicable to multiple FI techniques
  - i.e. exploitation is independent from injection

Exploitation is independent from injection!



- Attack works if the faults fits the chosen fault model
- Setup changes but the exploitation strategy stays the same

“FAULT ATTACKS ARE INJECTION DEPENDENT.”

"Glitch resolution is key to success"

- Shorter glitches definitely have advantages…
- But may not always be needed!

[2]Many sources mention removing decoupling capacitors for better result without giving a detailed reason. We were able to get voltage glitches to work both with and without removing the decoupling capacitors. It is our belief that removing the decoupling capacitors changes the response of the ringing and therefore the parameters for a successful glitch. But in our case, it does not make it any more or less tractable.

Yifan Lu – "Attacking Hardware AES with DFA" – (PS Vita)

Paper/Blog

Lesson learned: always try first…

GLITCH RESOLUTION IS KEY TO SUCCESS

“Synchronization with the target is required.”

- Synchronizing with target clock allows for increased precision.
- Often not possible.
  - Clock signal not reachable
- Our research is usually performed *without clock synchronization*
- Fast setup and short attack cycles increase attempts per second:
  - Speed overcomes target jitter

SYNCHRONIZATION WITH THE TARGET IS REQUIRED

DEBUNKED

“Successes rate determines attack feasibility”

- Fault attacks typically have a success rate < 100%

- Let’s assume two attacks, which one is more effective?
  - Attack A: 1% success rate, 10 attempts per minute
  - Attack B: 0,1% success rate, 1000 attempts per minute

- Success rate only provides fault frequency
  - Feasibility better described by “average time for success”

SUCCESS RATE DETERMINES ATTACK FEASIBILITY

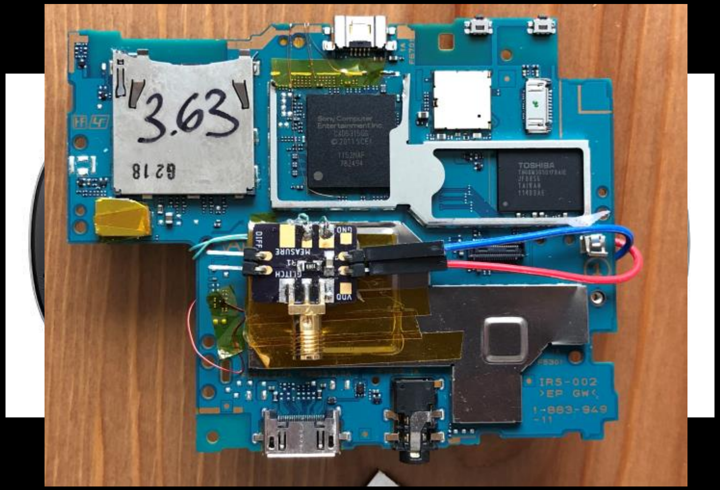
“Fault injection attacks do not scale.”

- They don’t. *Their results do.*
- Get assets out once and profit forever (e.g. code, keys, etc.).

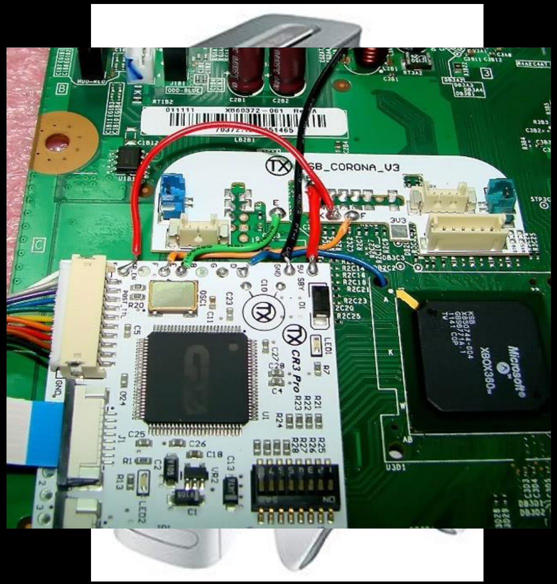
What do they have in common?

“Fault injection attacks do not scale.”

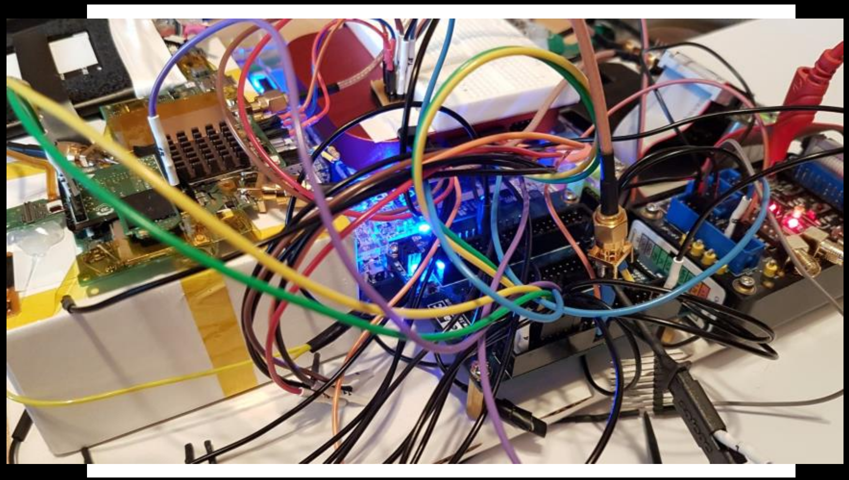
- They don’t. *Their results do.*
- Get assets out once and profit forever (e.g. code, keys, etc.).



Yifan Lu



Team Xecuter



Bernhard Froemel

Assets compromised using Fault Injection

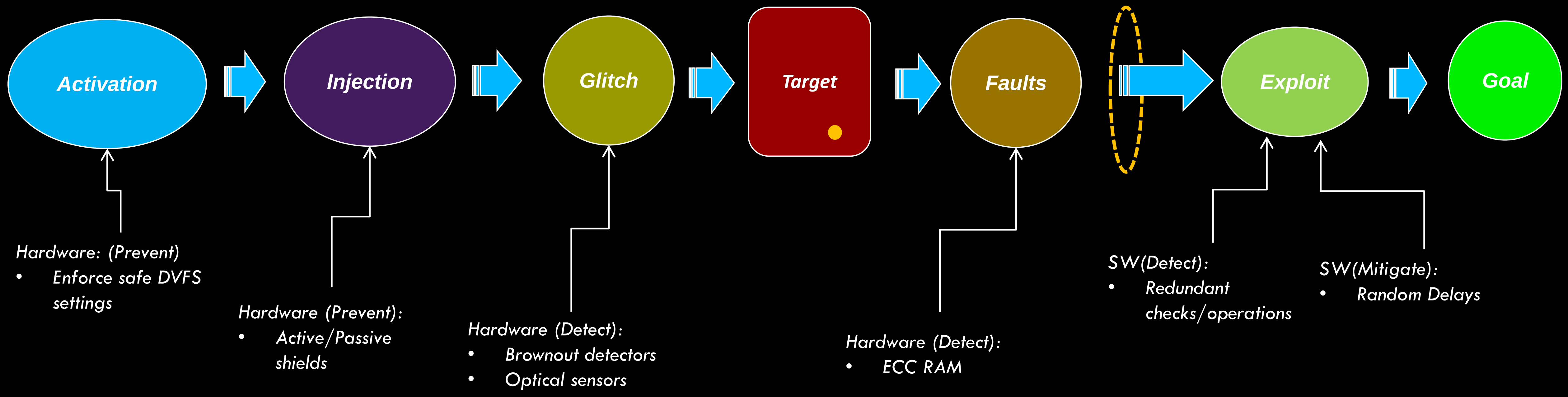
FAULT INJECTION ATTACKS DO NOT SCALE

"Implementing countermeasures is easy."

- How do you harden products against fault injection attacks?
  - *"Just add some random delays…"*
  - *"We have **triple** checks here. You CANNOT do it."*
  - *"We HAVE brownout detectors and clock monitors. Solved."*
  - *"There are NO CONDITIONALS to attack. It's SECURE!"*

Wait a minute…

Visualizing FI Countermeasures



Important

- Software countermeasures:
  - Specific to exploitation
  - Depend on selected fault model
  - Do not prevent/detect injection
- Hardware countermeasures:
  - CAN prevent injection
  - MAY be specific to injection technique

Systematic approach is essential to say something useful…

LET’S EXACTLY DO THAT

One Glitch, Multiple Faults…

Fault Attacks on Secure Embedded Software: Threats, Design, and Evaluation

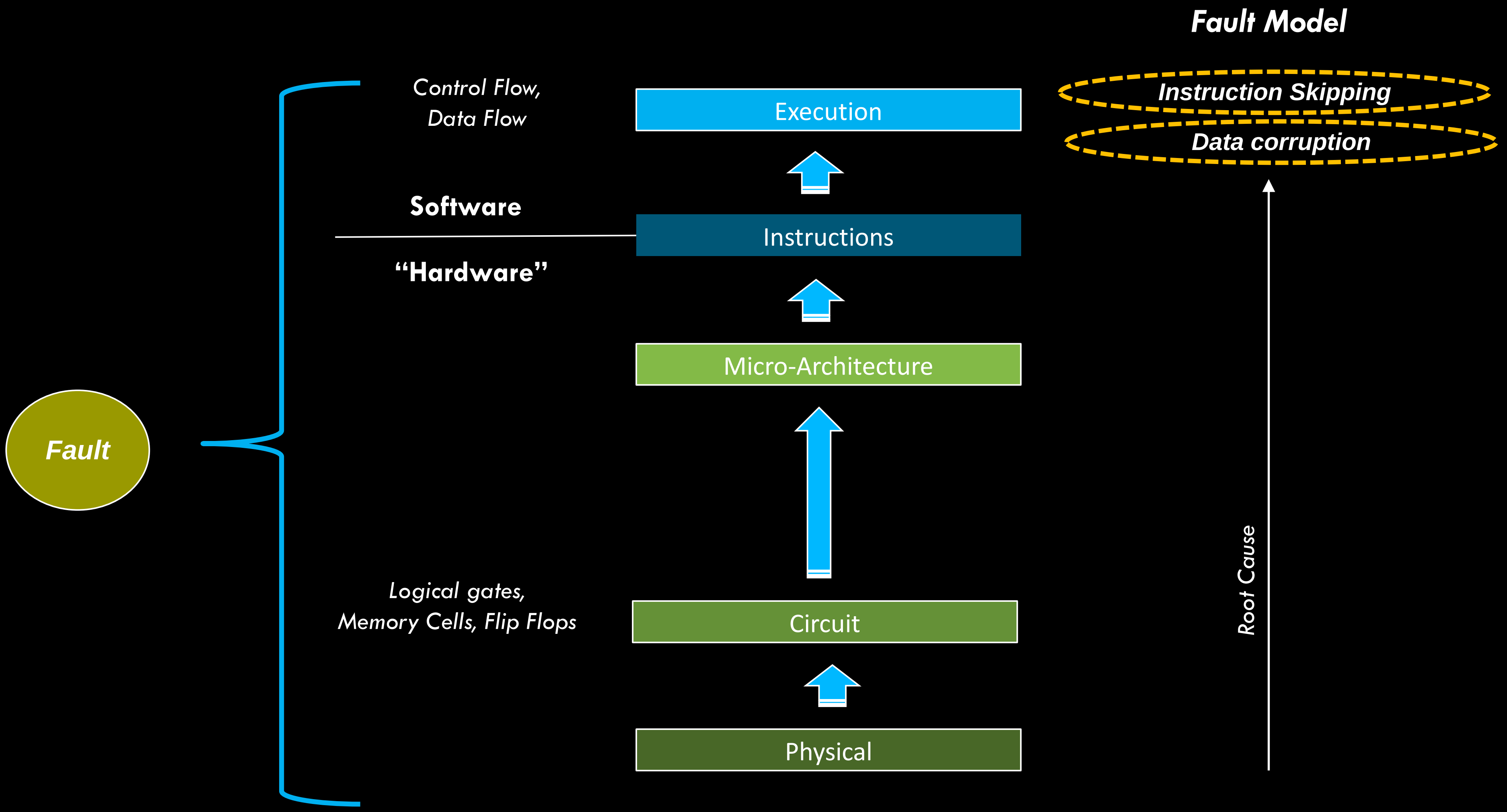
Bilgiday Yuce[1] · Patrick Schaumont[1] · Marc Witteman[2]

Received: 2 February 2018 / Accepted: 26 April 2018 / Published online: 10 May 2018

Abstract
Embedded software is developed under the assumption that hardware execution is always correct. Fault attacks break and exploit that assumption. Through the careful introduction of targeted faults, an adversary modifies the control flow or data flow integrity of software. The modified program execution is then analyzed and used as a source of information leakage, or as a mechanism for privilege escalation. Due to the increasing complexity of modern embedded systems, and due to the difficulty of guaranteeing correct hardware execution even under a weak adversary, fault attacks are a growing threat. For example, the assumption *that an adversary has to be close to the physical execution of software, in order to inject* an exploitable fault into hardware, has repeatedly been shown to be incorrect. This article is a review on hardware-based fault attacks on software, with emphasis on the context of embedded systems. We present a detailed discussion of the anatomy of a fault attack, and we make a review of fault attack evaluation techniques. The paper emphasizes the perspective from the attacker, rather than the perspective of countermeasure development. However, we emphasize that improvements to countermeasures often build on insight into the attacks.

Keywords Fault attacks · Secure embedded software · Embedded systems

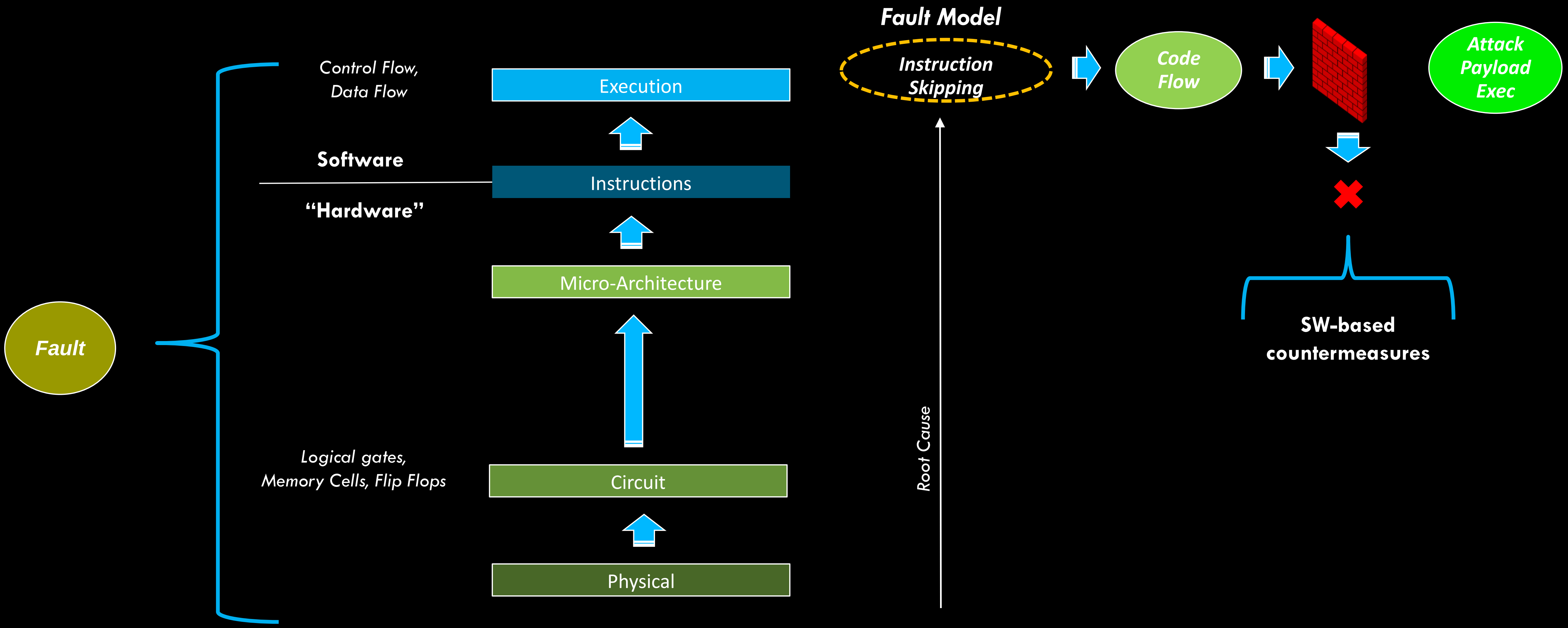
One Glitch, Multiple Faults



[2018]: Yuce, Schaumont, Witteman

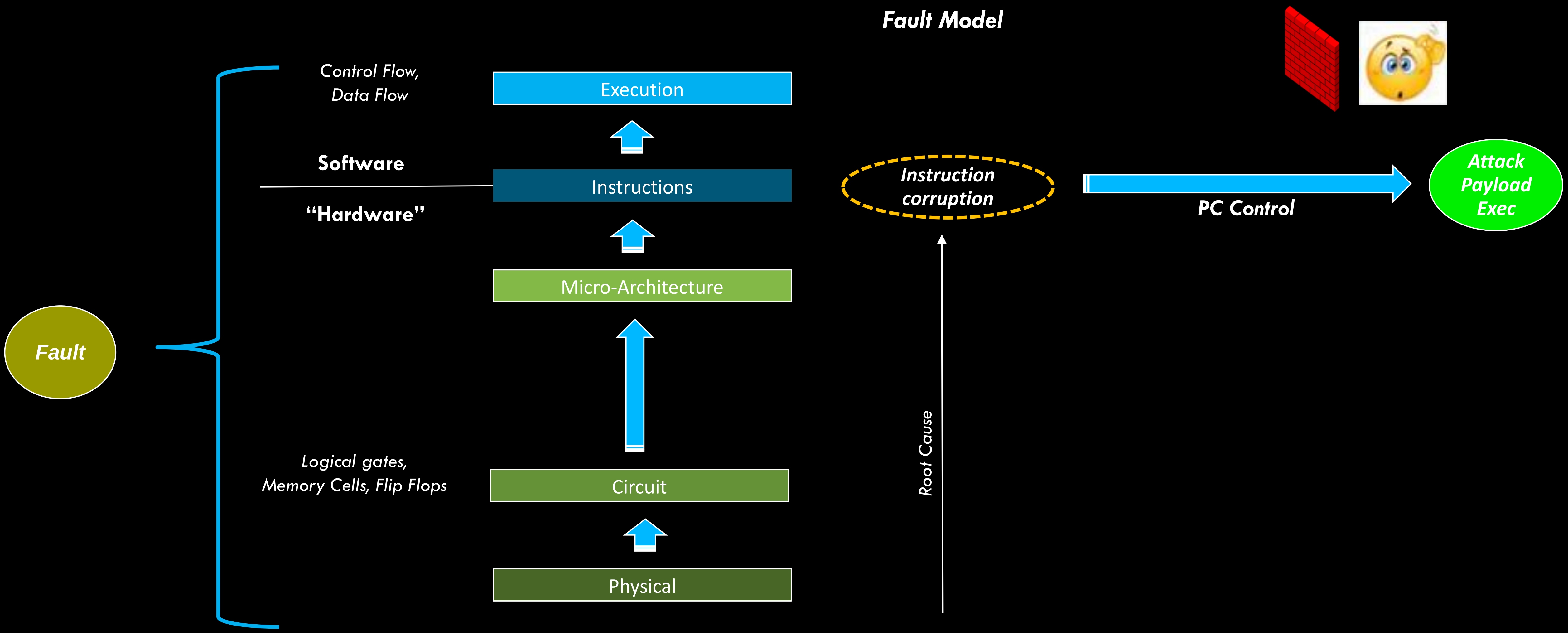
HARDENING SECURE BOOT

Secure Boot: Skipping Signature Check

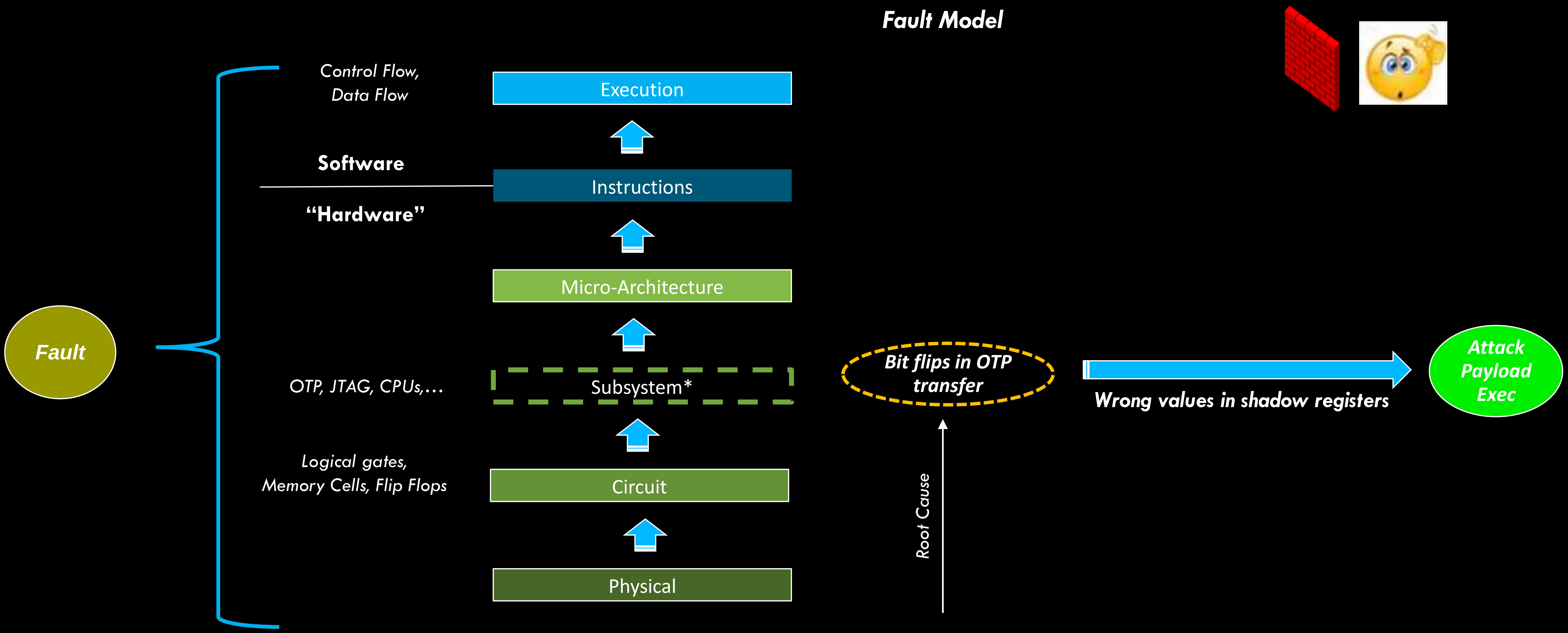


BUT…

Secure Boot: Instruction Corruption



Secure Boot: OTP Transfer Attack



*Extension to [2018]: Yuce, Schaumont, Witteman

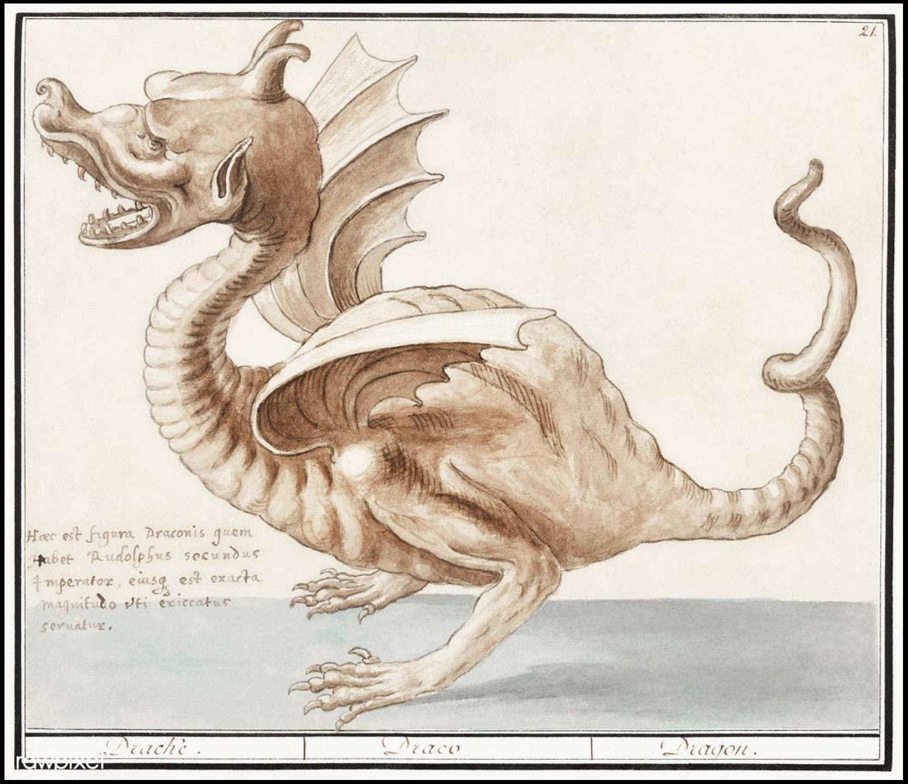
To summarize…

- Most SW countermeasures can be bypassed by:
  - Leveraging faults at a different system layer
- Countermeasures based on attack-specific assumptions
- Defenses CANNOT be implemented using software only
  - Fault injection hardened hardware is fundamental

IMPLEMENTING COUNTERMEASURES IS EASY

LET’S WRAP UP

Did we **REALLY** debunk all these myths?



“PLAUSIBLE DENIABILITY”, AT LEAST.

Takeaways

- Knowledge gaps between community, academia and industry.
  - Consolidation required to prevent incorrect conclusions.
- A common understanding will give ground to new and powerful FI attacks.
  - We hope this presentation helps with exactly that.
- Fault injection has reached the masses.
  - It is here to stay and will not go away.

HITB+CyberWeek
Abu Dhabi, UAE: 12-17 October 2019

TwentyTwo

Thank you!

PULSE

Niek Timmers
niek@twentytwosecurity.com
@tieknimmers

Cristofaro Mune
c.mune@pulse-sec.com
@pulsoid

Feel free to contact us!