SYSGO
EMBEDDING INNOVATIONS

TECHNOLOGY INNOVATIONS AT SYSGO
DDESC 2022

info@sysgo.com

THE SELECTED TOPICS

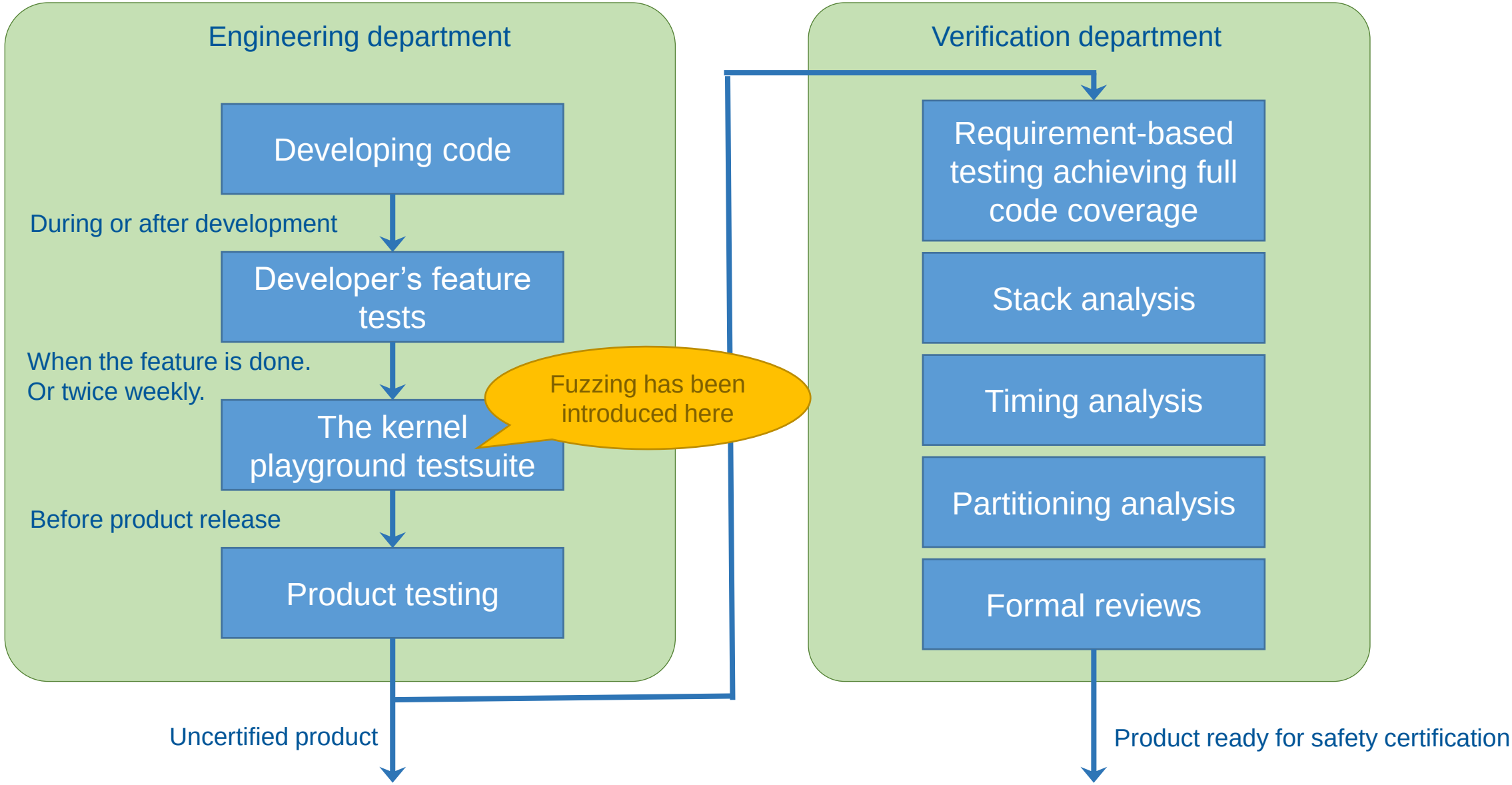
I. Fuzz testing and its benefit
II. Hardware-assisted intrusion detection system for RTOS
III. Certification for security and safety

Fuzz Testing and its Benefit

I. FUZZ TESTING – THE ENVIRONMENT

- PikeOS kernel is certified for safety for higher safety levels
  - For example, bugs in the kernel may cause hazardous conditions of an airplane
  - Therefore, robust verification processes and systematic testing are in place for many years
- Developers very competent professionals with habits of:
  - Attention to detail, testing their code on their own seriously, making mistakes only sparsely
  - They usually peer-review their code
  - As a safety net they have a further independent testsuite (playground) stressing the kernel for long time by very diverse means
- On top of this verification department performs verification processes independently:
  - Systematic testing
    - For example, PikeOS kernel has ~1900 interface requirements, each being tested
  - Many other activities and analyses

I. FUZZ TESTING – THE ENVIRONMENT

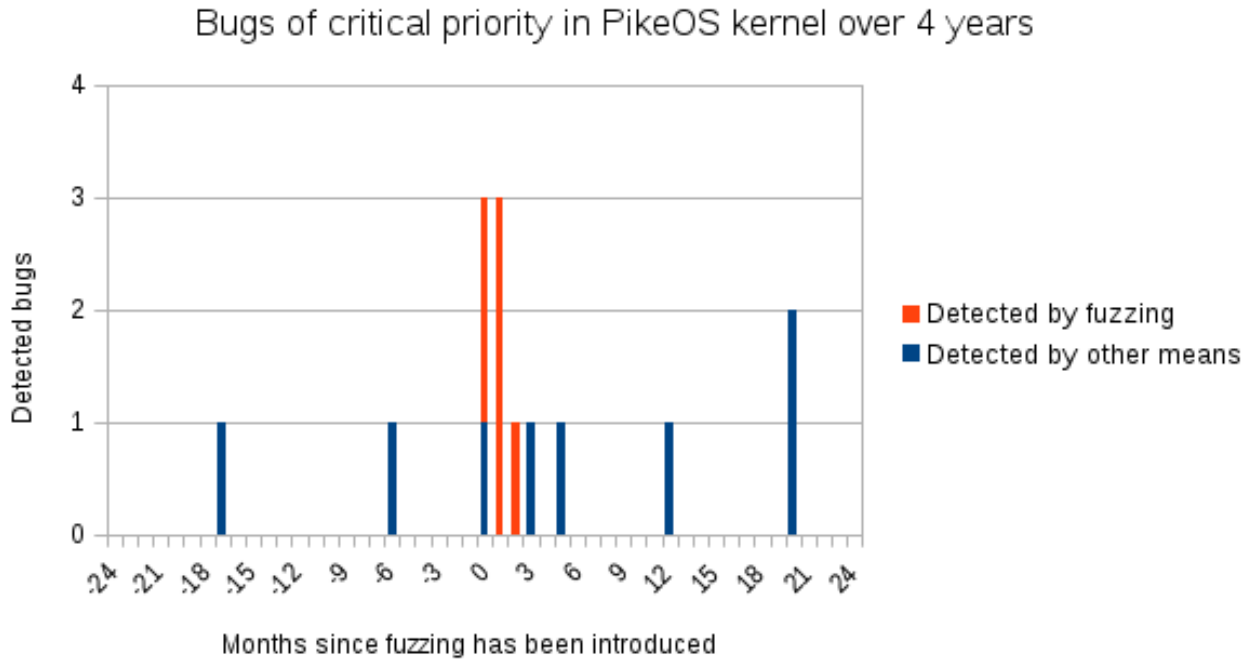


I. THE KERNEL PLAYGROUND TEST SUITE

- A long run/stress test suite extended by fuzzing:
  - Most syscalls wrapped, their arguments are randomized
  - These syscalls are called from a randomized hierarchy threads and tasks
  - A blocker task randomly blocking CPUs
  - An IPC task randomly issuing inter-process communication
  - CPU allocation randomized, priorities randomized, address space layout randomized, …
  - From fuzzing perspective too many interesting details to fit on one slide

- Currently:
  - Executed twice weekly for 4 kernel variants for 30 mins on 59 hardware platforms
  - The test suite detects 33-50% of bug reports that trigger a kernel assertion or a kernel panic

I. ILLUSTRATING THE BENEFIT OF FUZZ TESTING



- Focusing just on reported bugs of “critical” priority the fuzzer initially detected 6 such bugs
  - These bugs were strongly desired to get addressed soon
  - 5 of them were classified for possible safety consequences
  - Triggering them required complex, unusual and often multithreaded conditions
- After the initial period the benefit of fuzzing cannot be interpreted
  - fuzzer was integrated into early development phases, so its findings do not get into bug reports

- Conclusion: Fuzz testing was an efficient complement in testing approaches of PikeOS kernel

Hardware-assisted Intrusion Detection System for RTOS

II. HARDWARE-ASSISTED IDS FOR RTOS

- Utilizing ARM CoreSight to capture control-flow traces of monitored application
  - Control-flow traces ~ branches, jumps or other non-linear flow in the program execution
  - CoreSight stores the traces into a limited Embedded trace buffer
- IDS architecture (simplified):
  - Monitored application runs on a single core, the trace processing server on second core
  - The trace processing server suspends the application when the trace buffer gets full to process it
    - Thus, having impact on schedulability
    - A method how to construct feasible schedules was proposed
      - It introduces performance reduction
- Control-flow integrity check
  - Application “footprint” approach has been used
  - The footprint is the rate of trace buffer overfills per processing server period
    - Simple. Actually, with promising detection accuracy.
  - This may be extended by more sophisticated checks (see [1])
    - But these may cause further issues with scheduability and performance

[1] Towards Transparent Control-Flow Integrity in Safety-Critical Systems, ISC 2020

II. HARDWARE-ASSISTED IDS FOR RTOS

- Evaluation
  - TACLeBench benchmark, multimedia processing single-threaded applications
  - Footprint obtained during the training phase
  - Then, the application got modified by exploiting added stack overflow vulnerability
  - 100% detection accuracy for great majority of benchmark applications
- Details published in [1]

[1] Safety-Aware Integration of Hardware-Assisted Program Tracing in Mixed-Criticality Systems for Security Monitoring, RTAS 2021

II. HARDWARE-ASSISTED IDS FOR RTOS

- Strengths
  - Promising accuracy (to be verified on more diverse set of applications)
  - No internal knowledge or interaction of monitored application (e.g. instrumentation, …)
- Neutral properties
  - Typically observed detection time in two processing server periods
    - May not be sufficient to detect fast intrusions soon enough
- Weaknesses
  - The proposed scheduling framework may still be optimized to utilize multiple cores better
  - The monitored application is suspended during trace processing
    - Significant performance slowdown
      - May be bounded, but then not all traces get processed
      - If unbounded we experienced 40-605% CPU capacity needed for trace processing

Certification for Security and Safety

III. CERTIFICATION FOR SECURITY AND SAFETY

- Goals of our efforts:
  - Introduce more precedents of embedded RTOS usages that are certified for security in safety context in order to help establishing common practices:
    - The standards for securing safety-critical embedded systems are not yet fully developed
    - Regulations for securing critical infrastructure are not yet fully designed and required
    - Markets are sometimes hesitant to certify industrial embedded systems for security, partly because there is not yet much established practice
  - What safety certification artefacts can be reused for security certification?
  - How differing security standards relate to each other?
    - For example, interchangeability, evaluation processes, organizations involved, …
  - As PikeOS is a software component what is the methodology for compositional certification?
    - For example, IEC 62443 is well structured for composition of components, CC not so much

- More futuristic topic
  - Certifying AI for safety or security

III. CERTIFICATION FOR SECURITY AND SAFETY

- Outcomes:
  - We applied PikeOS in railway, subway and smart grid while having it certified for IEC 62443 as a Multiple Independent Layers of Security (MILS) system
    - PikeOS is certified for Common Criteria only

    [1] "we conclude that a CC certification of a separation kernel suffices for use as subcomponent of a product under 62443-4-1/62443-4-2 certification"

  - Reuse of safety certification artefacts and processes for security certification
    - Requirement database and tracing reusable or extendable
      - Security additions needed: threat modelling, tests for security aspects, penetration testing, user manual for maintaining the security properties, …
    - Safety verification is focused on documented API, because use of undocumented API is forbidden to application developers. This cannot be forbidden to hackers.

[1] Security Certification of Cyber Physical Systems for Critical Infrastructure based on the Compositional MILS Architecture, IECON 2021

QUESTIONS OR COMMENTS?

SYSGO GmbH

Am Pfaffenstein 8
55270 Klein-Winternheim
Germany

Phone: +49 6136 99480
E-Mail: info@sysgo.com

Subscribe, Like and Follow:

www.sysgo.com/newsletter

www.sysgo.com/twitter

www.sysgo.com/linkedin

www.sysgo.com/youtube

www.sysgo.com