WHERE INNOVATION BEGINS
DESIGN AUTOMATION CONFERENCE

emproof

Formal Verification Methods for Binary Rewriting

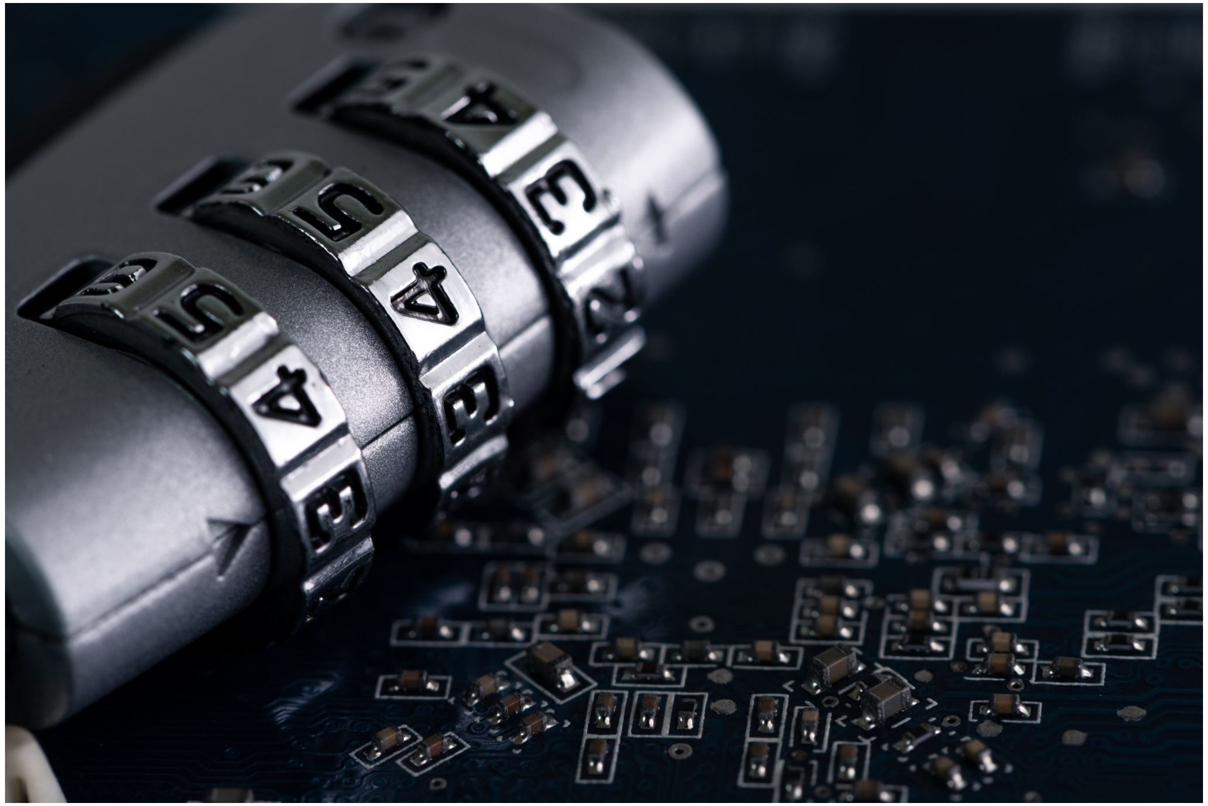
Tim Blazytko

About Me

- co-founder of Emproof B.V.
- binary security researcher
- PhD in software security
- automated program analysis
- formal verification

Goals: Securing Embedded Devices

- rewriting embedded firmware
- exploit mitigations
- IP protection
- support for various ISAs



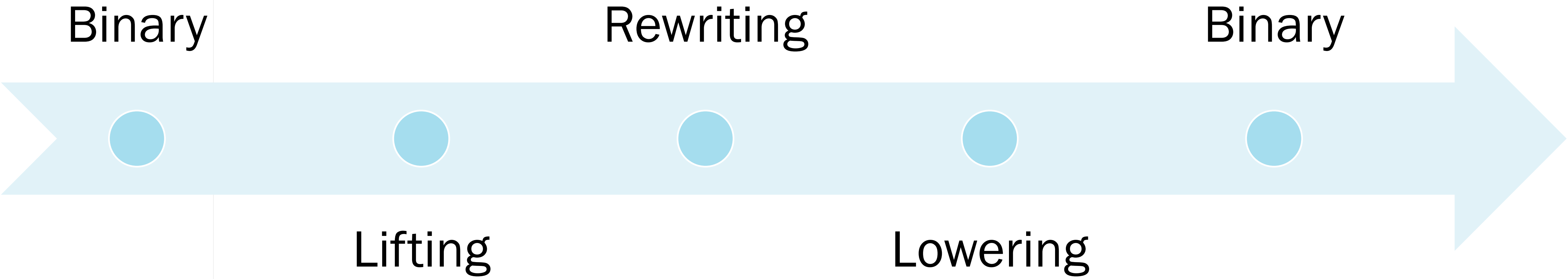
Exploit Mitigations

- detection of memory corruptions
- stack canaries
- control-flow integrity

IP Protection

- complicate reverse engineering
- code obfuscation
- anti-debug
- anti-tamper

Binary Rewriting



Binary

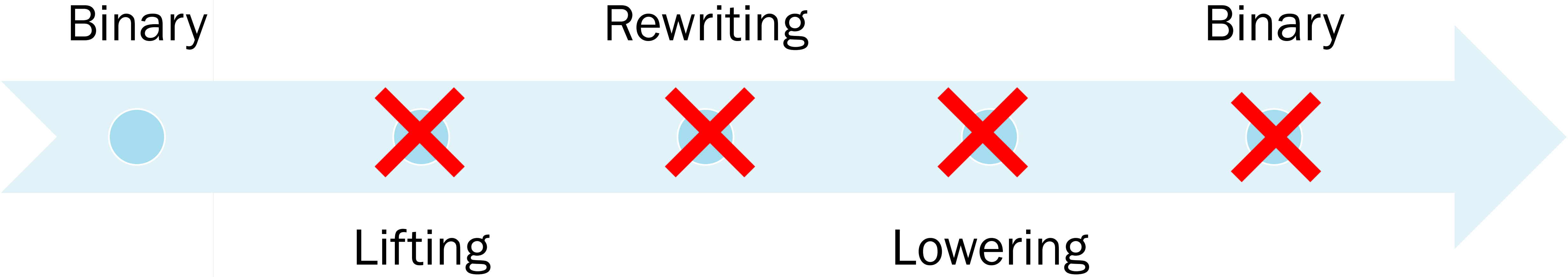
Lifting

Rewriting

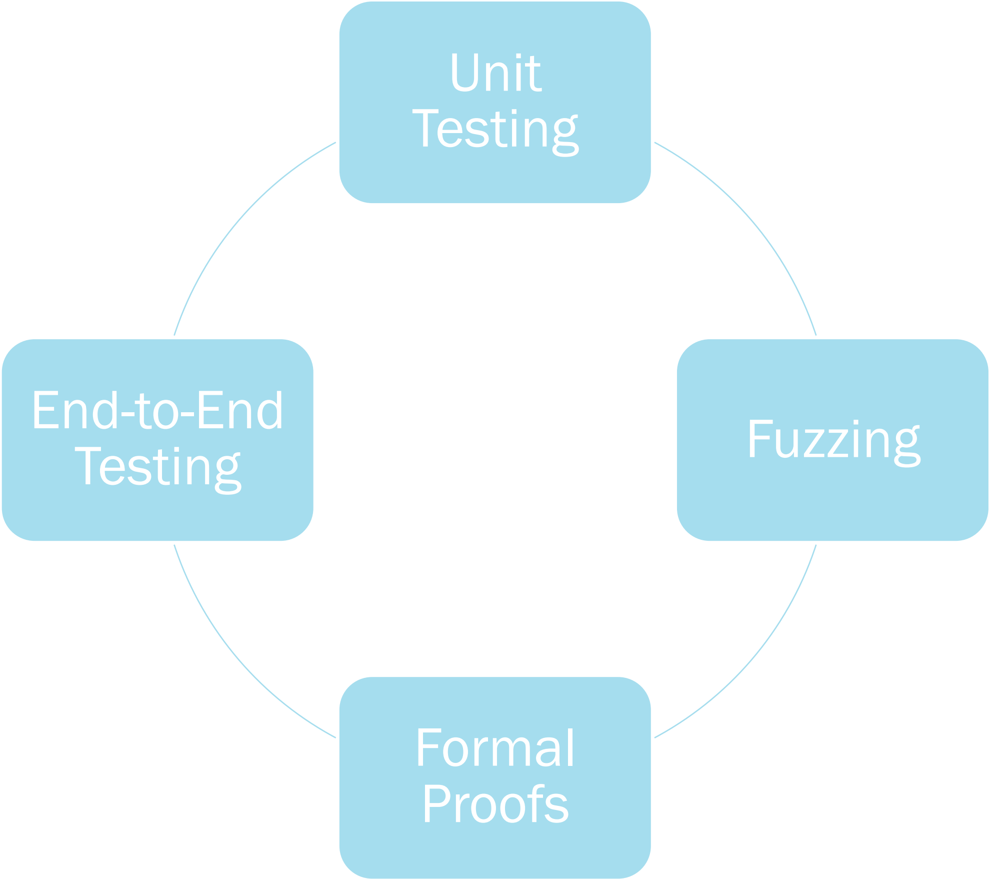
Lowering

Binary

Things Break Everywhere



Validation & Verification Life Cycle



Unit Testing of Components

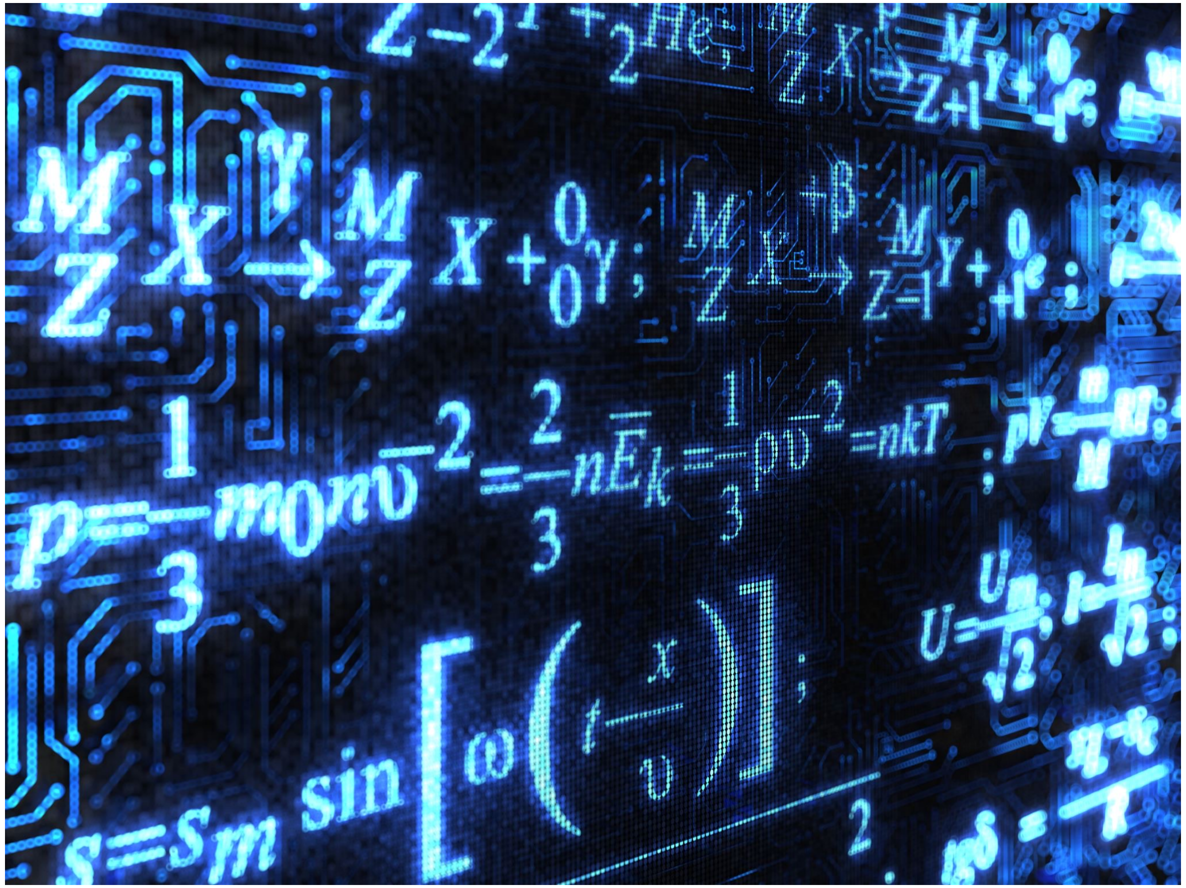
- clear specification of (in)valid behavior
- new bugfix ⟶ new unit test
- hundreds of unit tests

Fuzzer for Individual Components

- random I/O testing to break stuff
- domain knowledge to craft inputs
- component specific fuzzers

Formal Proofs

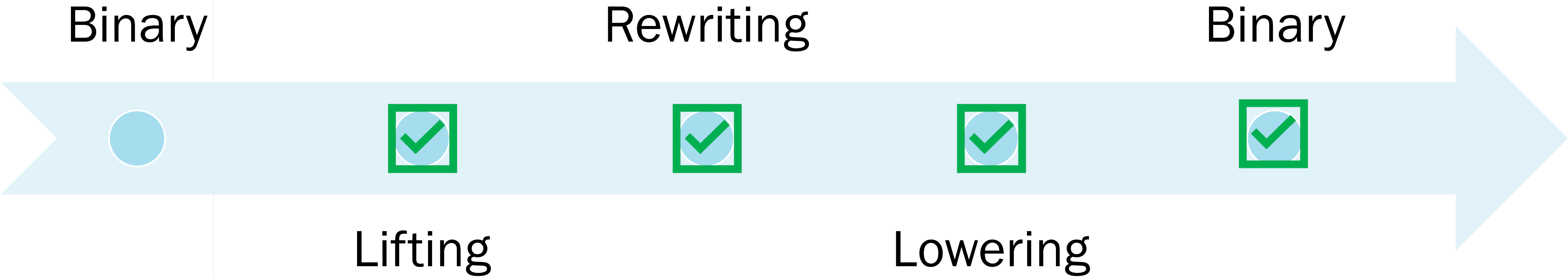
- validation of assumptions specific to code transformations
- SMT solvers to prove semantic equivalence
- applied after each transformation



End-to-End Testing

- large-scale end-to-end testing
- various configurations & binaries
- embedded cloud infrastructure
- original vs. modified binary

CI/CD-based Validation & Verification



Binary

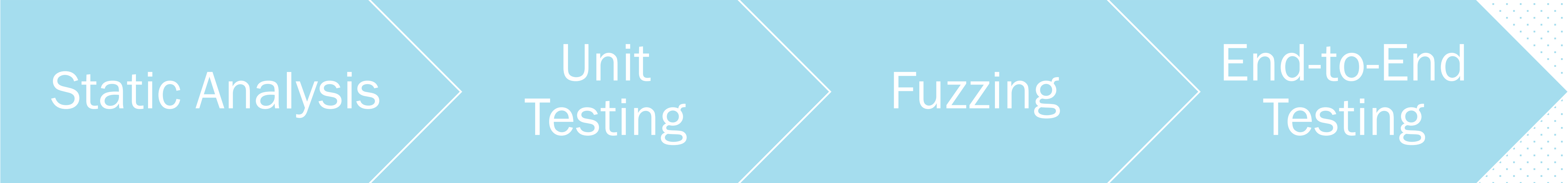
Lifting

Rewriting

Lowering

Binary

CI/CD Infrastructure (Automation)



Static Analysis

Unit Testing

Fuzzing

End-to-End Testing

Conclusion

- binary rewriting to secure embedded devices
- exploit mitigations & IP protection
- validation & verification on all levels
- contact: tblazytko@emproof.com