9th UCAAT User Conference on Advanced Automated Testing

Towards the Absence of Bugs

An Intelligent Combination of Static and Dynamic Analysis to Identify Vulnerabilities in the Development Process

Ramon Barakat

Fraunhofer FOKUS

15/09/2022

Agenda

Tool-supported Security Testing today

- Static and dynamic analysis
- Interactive analysis

Verification of static analysis findings

Residual Risk Estimation

- Good-Turing Estimator (GTE)

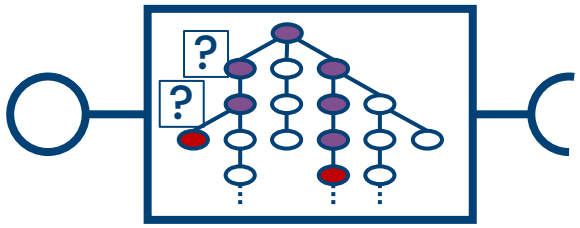
Conclusion

Acknowledgments
This work was developed within the IntelliSecTest project. The IntelliSecTest project is funded with 3.5 million euros over three years by the Fraunhofer PREPARE program.

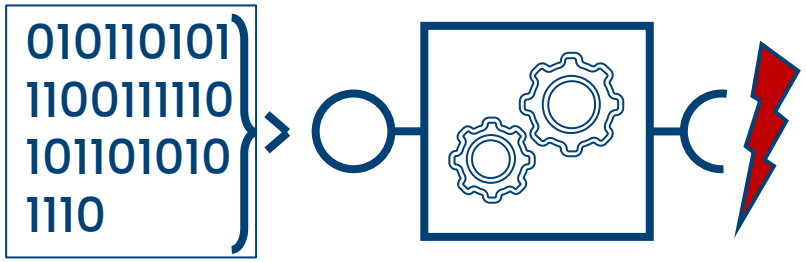
Tool-supported Security Testing today

Two major approaches

Static Analysis



Dynamic Analysis

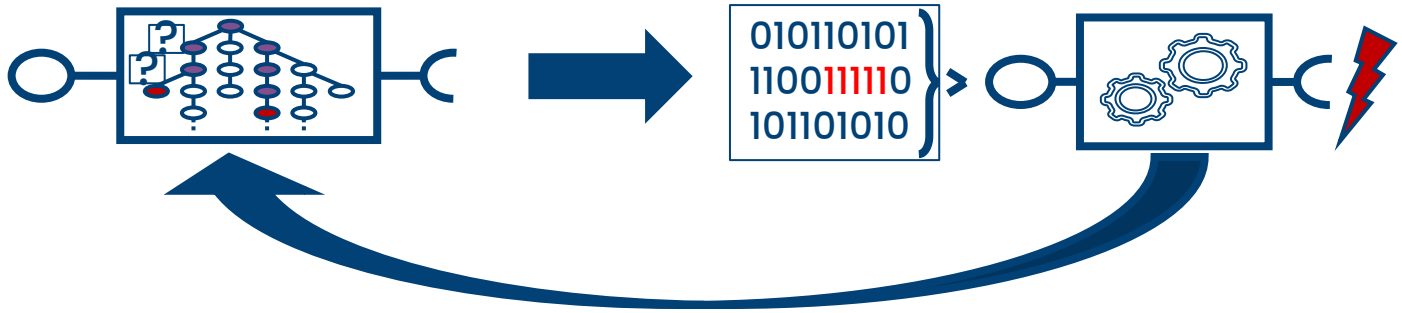


	Static Analysis	Dynamic Analysis
Advantages	▪ High path coverage ▪ Good presentation of results	▪ Very few false positives ▪ Provides input data triggering to vulnerability
Drawbacks	▪ High number of false positives	▪ Random path coverage ▪ Poor results presentation

Tool-supported Security Testing today

Interactive Analysis

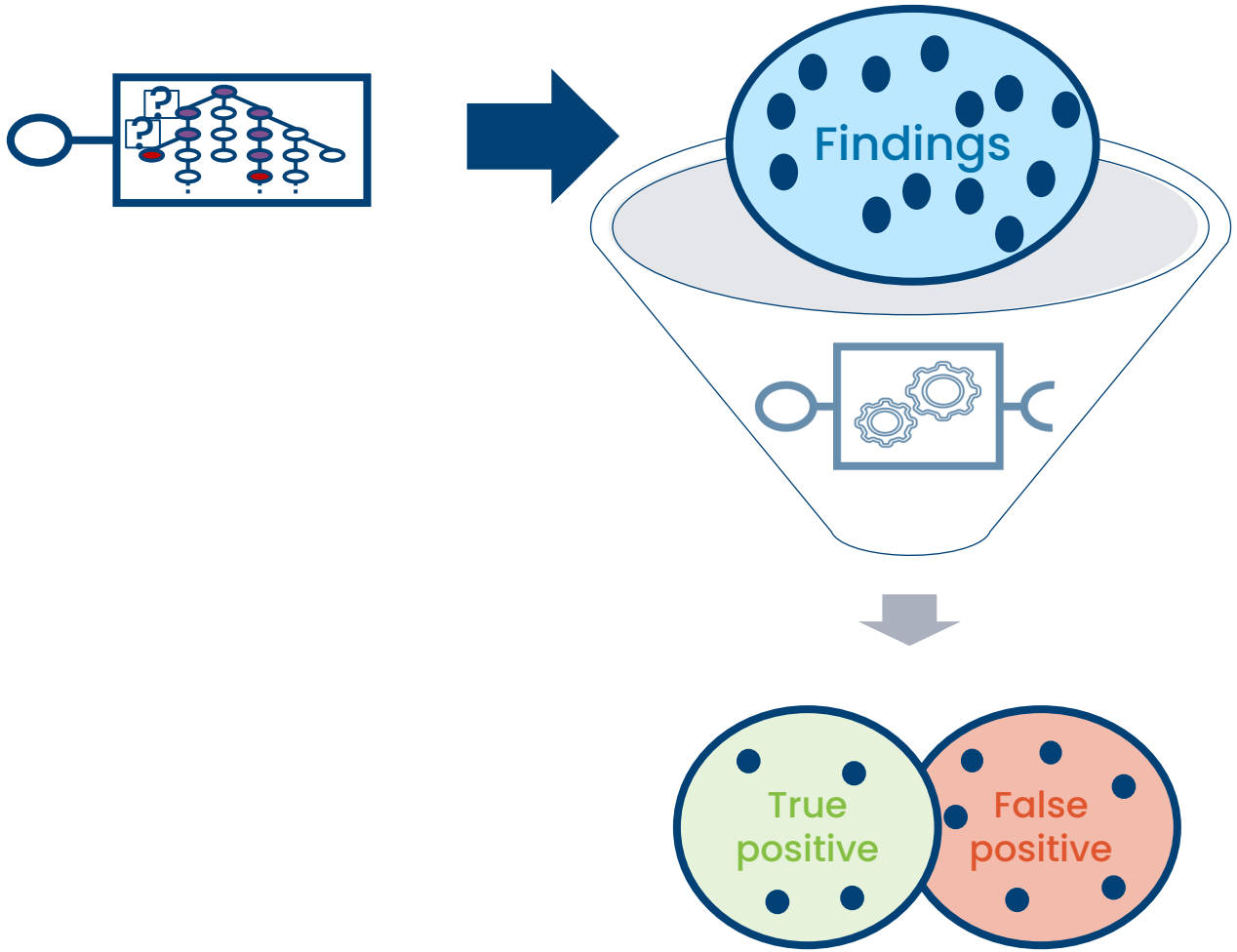
Combination of static and dynamic Analysis



Advantages

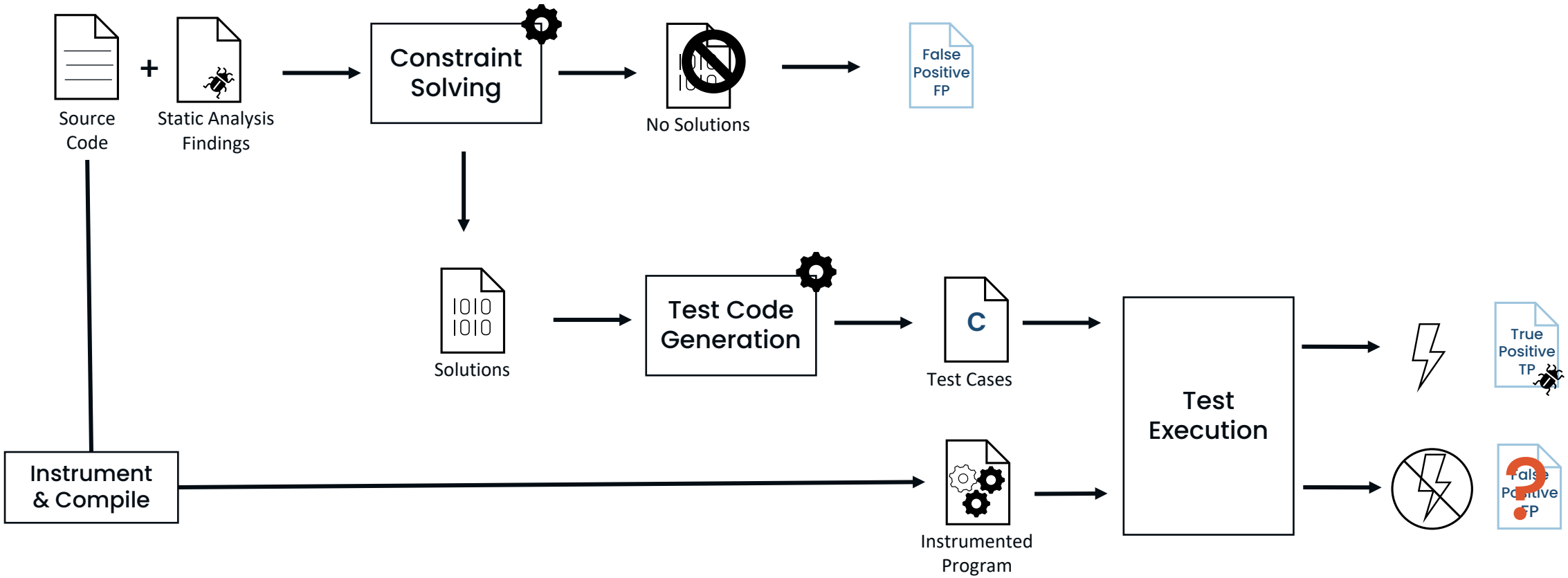
- High path coverage
- Good presentation of results
- Provides input data leading to vulnerability
- Reduce number of false positives

Verify static analysis findings

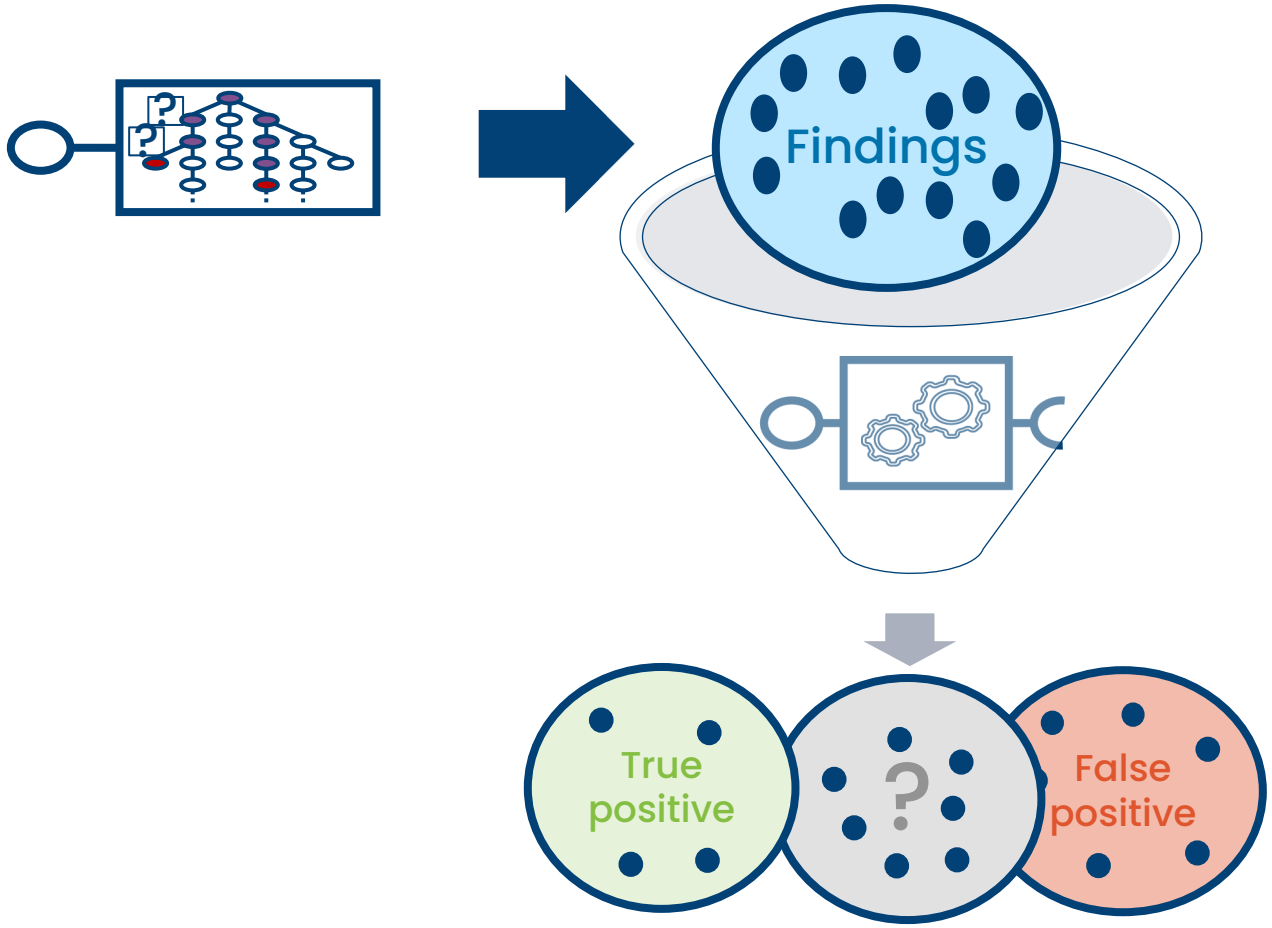


Verify static analysis findings

Create test cases using constraint solving



Verify static analysis findings



> *"Program* testing can *be used* to show the presence *of* bugs, *but never* to show their absence!"
>
> [E. Dijkstra]

→ Need a way to estimate the residual risk that there is an undiscovered vulnerability in the code

Residual Risk Estimation

Traditionally probability calculation:

- Assumption: the ratio of every element in the set in relation to the occurrence in the sample set is universally true
- Result: no prediction for unseen elements

Good-Turing Estimation (GTE):

- Assumption: the sample data just captures a part of the set
- Consequents: probability discounting to create room for unseen elements (pseudo count)

Residual Risk Estimation

Missing mass estimation

"the chance that the next […] sampled will belong to a new species is approximately"

$$P'_0 \approx \frac{n_1}{N}$$

P'_0 the probability for all unobserved species ("missing mass")
n_r number of species that were seen exactly r times
N is the total number of counts

Hence also the expected total chance of all species that are represented r times or more in the sample is approximately

$$N^{-1}\{(r+1)n_{r+1}+(r+2)n_{r+2}+\ldots\}. \quad (7)$$

In particular, the expected total chance of all species represented at all in the sample is approximately

$$N^{-1}(2n_2+3n_3+\ldots) = 1-n_1/N. \quad (8)$$

We may say that the proportion of the population represented by the sample is approximately $1-n_1/N$, and *the chance that the next animal sampled will belong to a new species is approximately*

$$n_1/N. \quad (9)$$

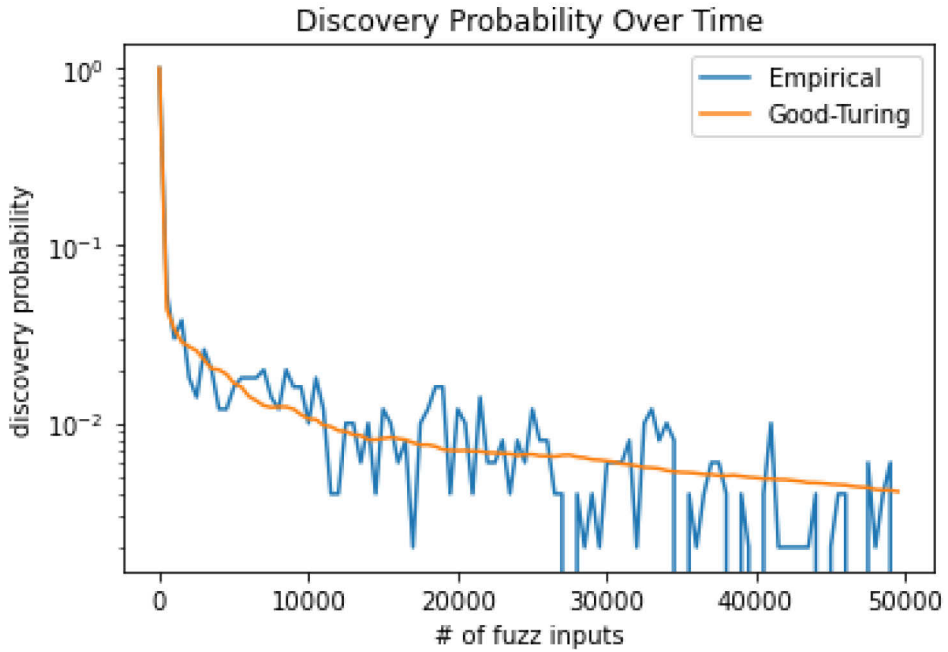
I.J.Good: THE POPULATION FREQUENCIES OF SPECIES AND THE ESTIMATION OF POPULATION PARAMETERS (1953)

Residual Risk Estimation

GTE Applying to Fuzzing

*"If no error has been exposed throughout the [fuzzing] campaign, the **Good-Turing estimator gives an upper bound** on the probability to generate a test input that exposes an error."*

[M. Böhme]



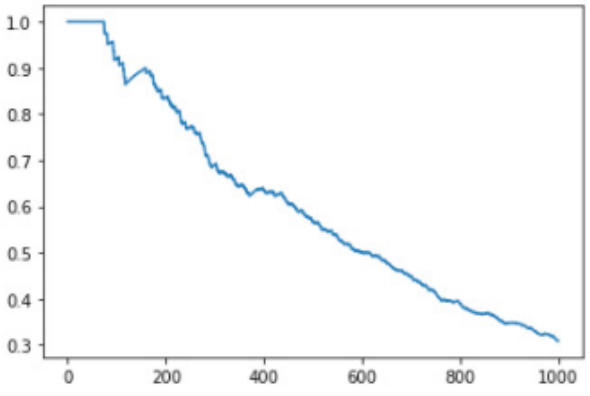
Empirical estimator

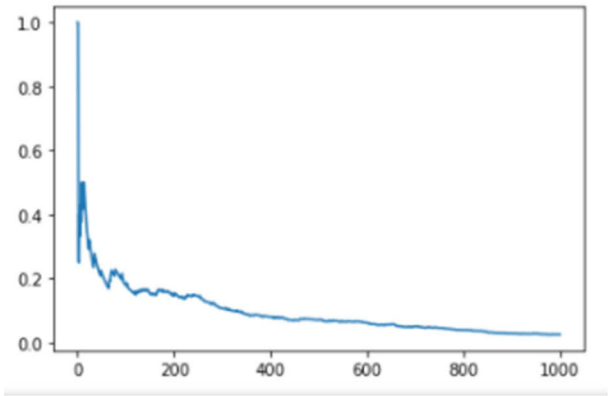
To measure the empirical probability, we execute the same population of inputs (n=50000) and measure in regular intervals (measurements=100 intervals). During each measurement, we repeat the following experiment repeats=500 times, reporting the average: If the next input yields a new trace, return 1, otherwise return 0. Note that during these repetitions, we do not record the newly discovered traces as observed. [1]

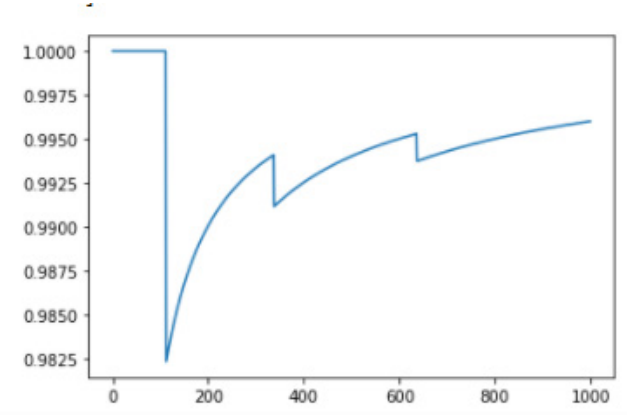
[1] https://www.fuzzingbook.org/html/WhenToStopFuzzing.html

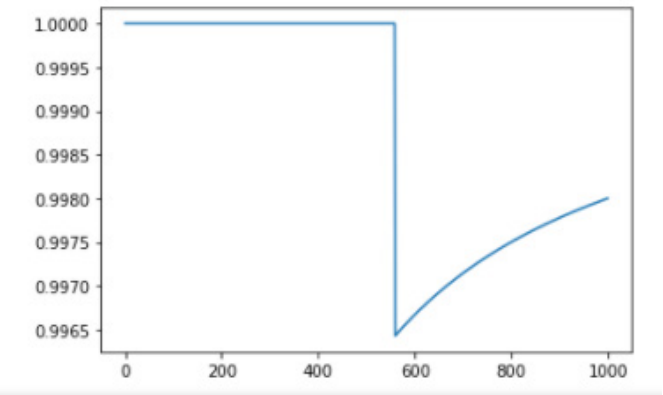
Residual Risk Estimation

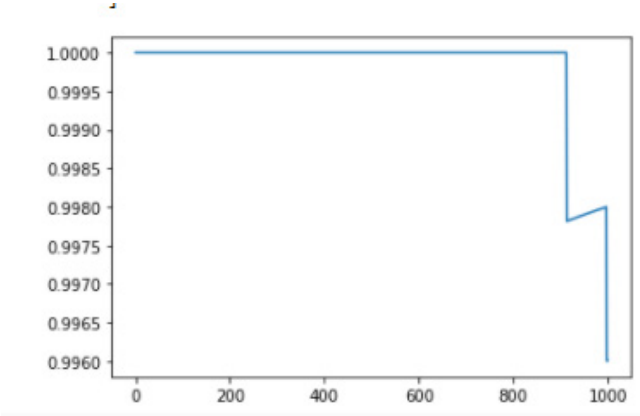
GTE Applying to different examples

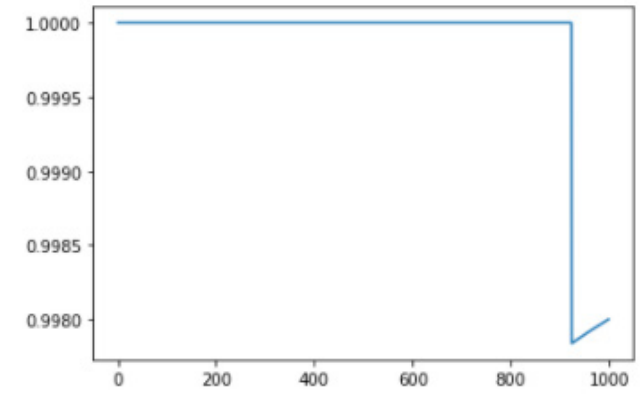






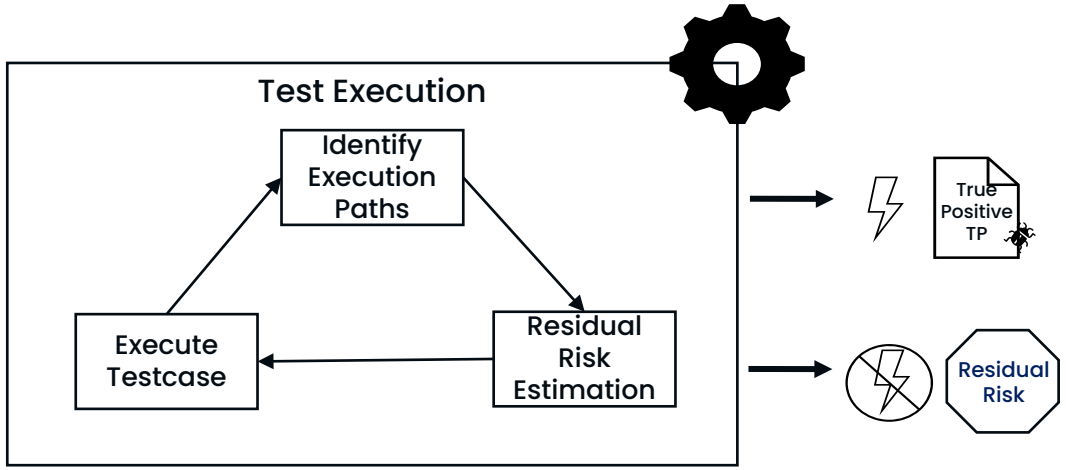






Residual Risk Estimation

Verification of Static Analysis Findings

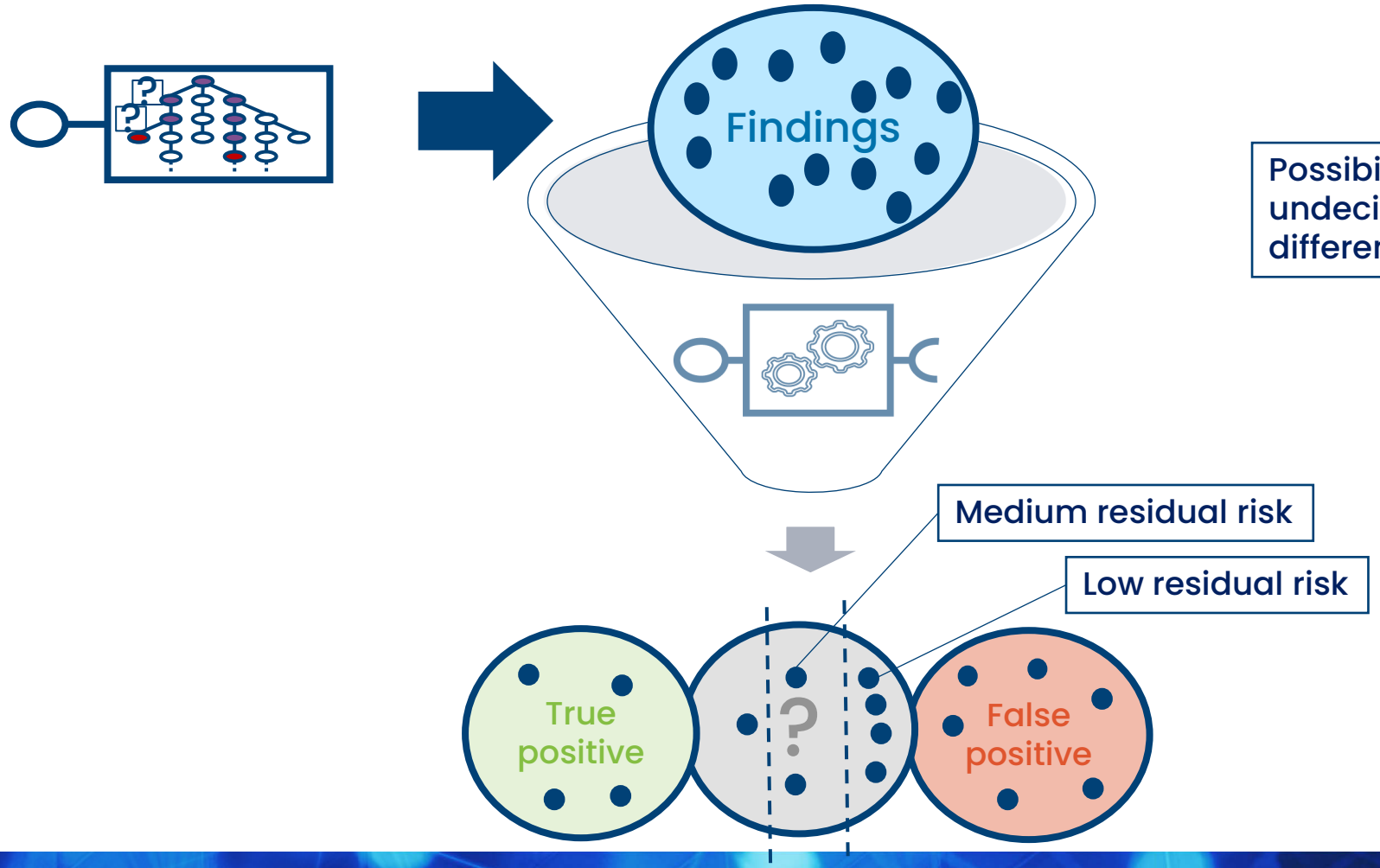


When to stop the test execution?

- Use relative GTE Values (no absolute value)
- Consider a calibration period
- Monitoring the trend in GTE values across multiple test cases
- End test execution when no more significant changes are monitored

Residual Risk Estimation

Verification of Static Analysis Findings



Possibility of grouping undecidable findings into different residual risk classes

Summary

- Static and dynamic analysis can benefit from each other
- Dynamic analysis can be used to verify static analysis findings
- Good-Turing estimation can estimate the residual risk of a test campaign

Even if tests does not provide absolute evidence,
a measure of evaluation can be provided
to reduce the degree of uncertainty

Thank You!

Any further questions?

Contact me:
ramon.barakat@fokus.fraunhofer.de

References

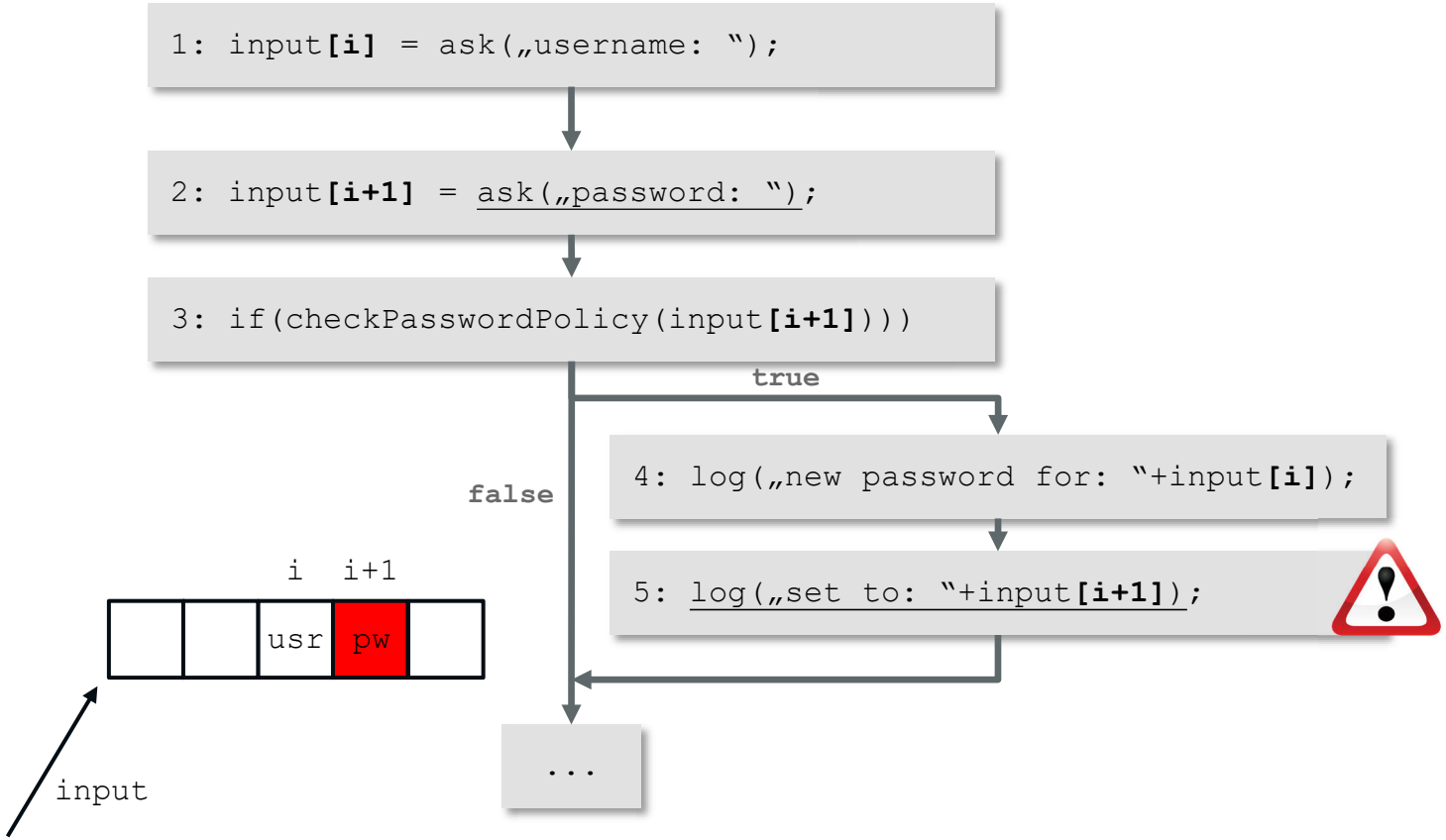
- Good, Irving J. "The population frequencies of species and the estimation of population parameters." *Biometrika* 40.3-4 (1953): 237-264.
- Böhme, Marcel. "STADS: Software testing as species discovery." *ACM Transactions on Software Engineering and Methodology (TOSEM)* 27.2 (2018): 1-52.
- https://www.fuzzingbook.org/html/WhenToStopFuzzing.html
- Gale, William A., and Geoffrey Sampson. "Good-turing frequency estimation without tears." *Journal of quantitative linguistics* 2.3 (1995): 217-237.
- Can Good-Turing Frequency Estimation Tell Us When to Stop Fuzzing? (Blog) https://bshastry.github.io/2018/10/08/good-turing-fuzzing.html

Tool-supported Security Testing today

Example: Implementation of a password change

Password leak in line 5!

(Password from line 2)



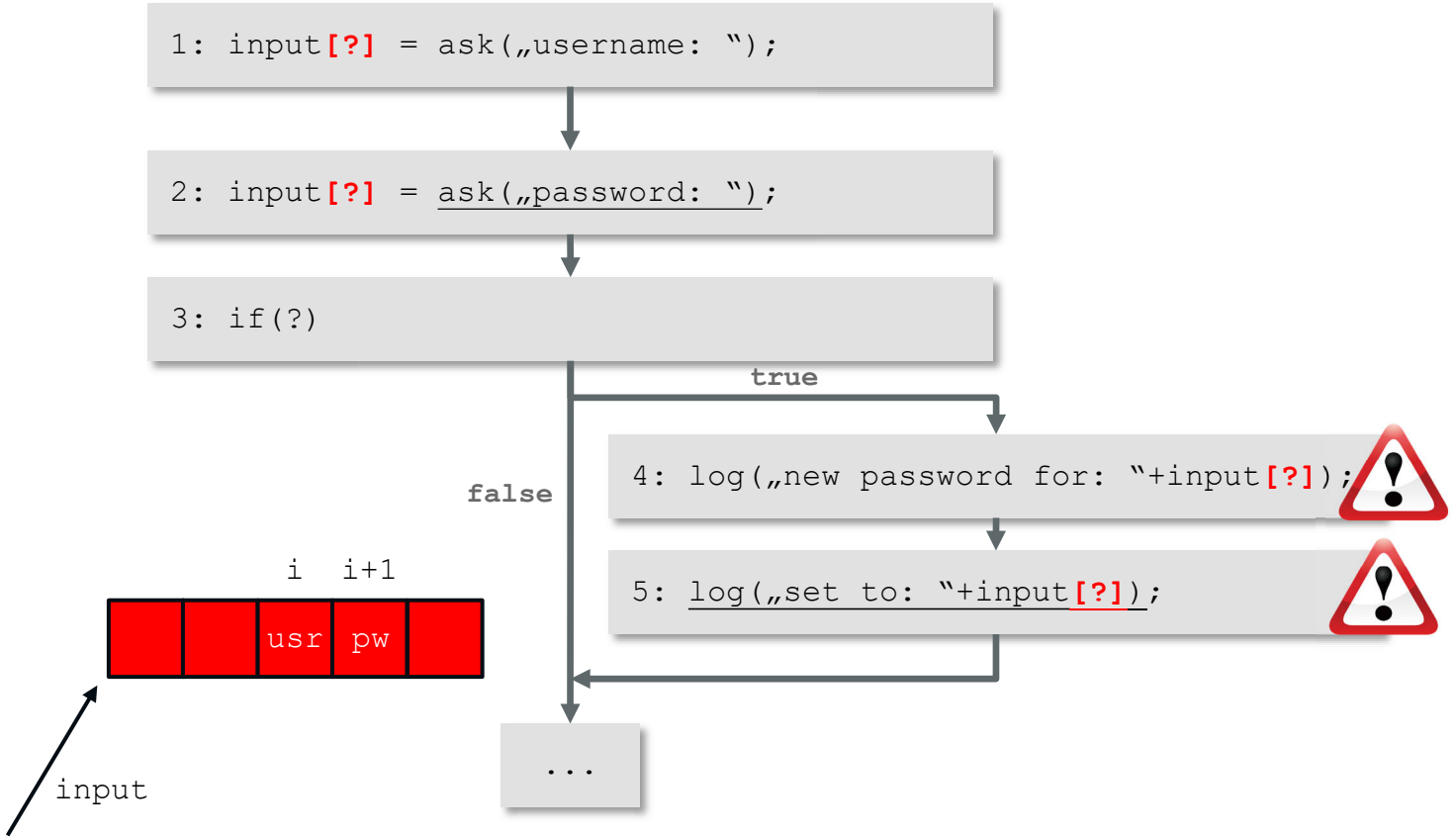
Tool-supported Security Testing today

Static Analysis

Common approximation:

- Abstracting path constraints
- Abstraction of array indices

➡ Wrong warning in line 4



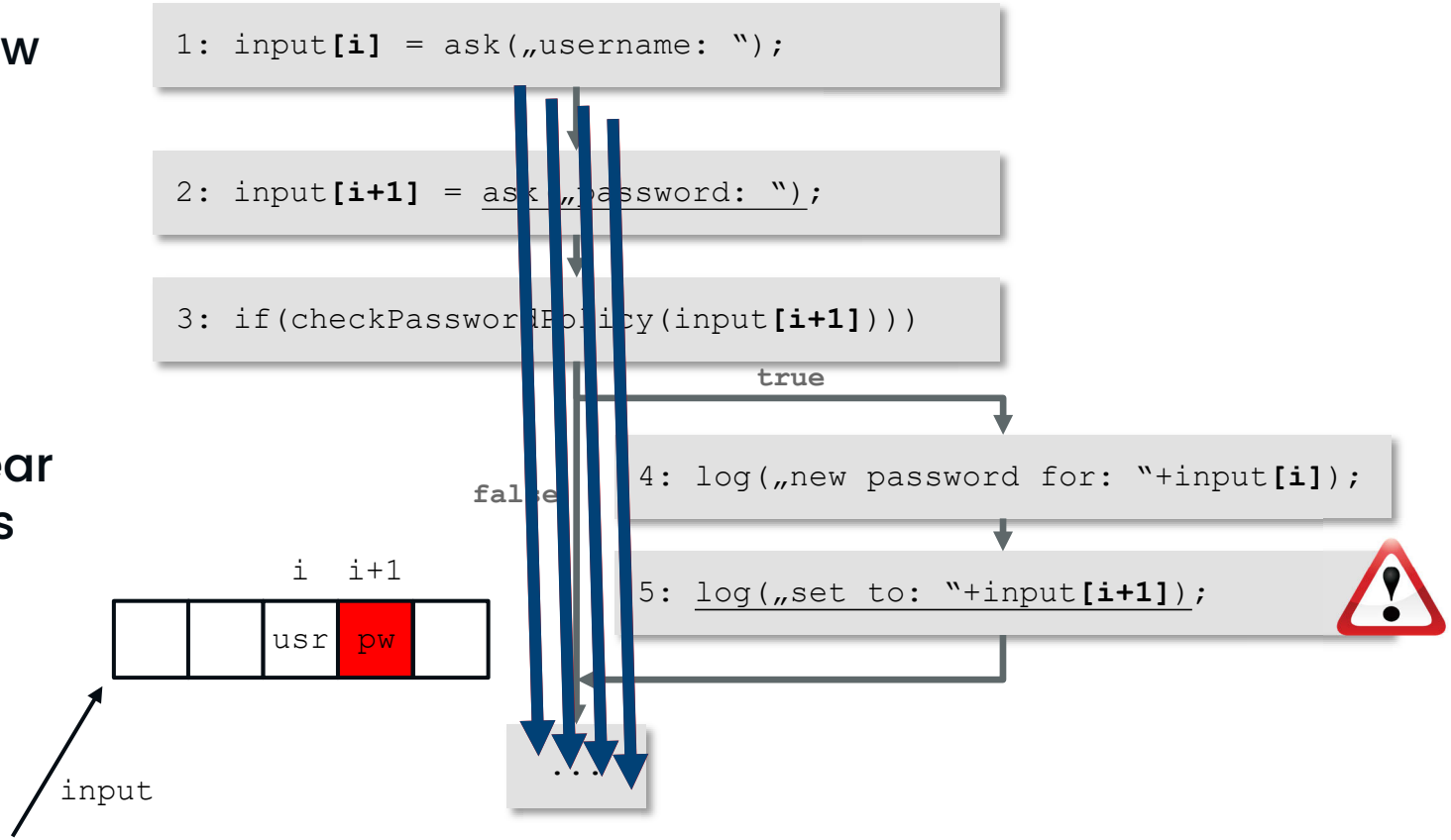
Tool-supported Security Testing today

Dynamic Analysis

- Probability to execute line 5 very low
- Vulnerability remains undetected

AND

- If password leak is observed, unclear where the vulnerable line of code is



Good-Turing Estimator

How does it work

GT Estimation:

$$\mathrm{P'_r} = \frac{1}{N}(r+1)\frac{n_{r+1}}{n_r}$$

- P'_0 the probability for all unobserved species ("missing mass")
- P'_r the probability to observe r individuals for species X
- r number of individuals that have been observed for species X
- n_r number of species that were seen exactly r times
- N is the total number of counts

Good-Turing Estimator

How does it work

Set: {a, b, c, d, e, f, g}

Sample data: "aabdeeefff" *(N = 10)*

GT Estimation: $P'_r = \frac{1}{N}(r+1)\frac{n_{r+1}}{n_r}$

- a = 2 times → r = 2
- b = 1 time → r = 1
- c = 0 times → r = 0
- d = 1 time → r = 1
- e = 3 times → r = 3
- f = 3 times → r = 3
- g = 0 times → r = 0

$n_0 = 2$

$n_1 = 2$

$n_2 = 1$

$n_3 = 2$

$P'(c,g) = P'_0 = \frac{1}{10} * \frac{2}{2} = \frac{1}{10}$

$P'(b,d) = P'_1 = \frac{2}{10} * \frac{1}{2} = \frac{1}{10}$

$P'(a) = P'_2 = \frac{3}{10} * \frac{2}{1} = \frac{6}{10}$

$P'(e,f) = P'_3 = \frac{4}{10} * \frac{?}{3} = \frac{?}{10}$

Use interpolation for higher counts or lacks