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
Tool-supported Security Testing today

Two major approaches

Static Analysis

- High path coverage
- Good presentation of results

Dynamic Analysis

- Very few false positives
- Provides input data triggering to vulnerability

Advantages

- High number of false positives

Drawbacks

- 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

- **True positive**
- **False positive**
Verify static analysis findings

Create test cases using constraint solving

- Source Code
- Static Analysis Findings
- Constraint Solving
- No Solutions
- False Positive FP
- Test Code Generation
- Solutions
- Test Cases
- Instrumented Program
- Test Execution
- True Positive TP
- False Positive FP
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

Findings

True positive

False positive

False negative
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”

\[ P'_{0} \approx \frac{n_{1}}{N} \]

\[ P'_{0} \quad \text{the probability for all unobserved species ("missing mass")} \]
\[ n_{r} \quad \text{number of species that were seen exactly r times} \]
\[ N \quad \text{is the total number of counts} \]

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]

Residual Risk Estimation

GTE Applying to different examples
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

- True positive
- False positive
- Medium residual risk
- Low residual risk
- False positive
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

Tool-supported Security Testing today

Example: Implementation of a password change

Password leak in line 5!
(Password from line 2)

```plaintext
1: input[i] = ask("username: ");

2: input[i+1] = ask("password: ");

3: if(checkPasswordPolicy(input[i+1])))

4: log("new password for: "+input[i]);

5: log("set to: "+input[i+1]);
```
Tool-supported Security Testing today

Static Analysis

Common approximation:
- Abstracting path constraints
- Abstraction of array indices

Wrong warning in line 4

```
1: input[?] = ask("username: ");
2: input[?] = ask("password: ");
3: if(?)
   true
   4: log("new password for: "+input[?]);
   false
   5: log("set to: "+input[?]);
```
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:

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

- \( P'_0 \): 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: \[ P'_r = \frac{1}{N} \left( r + 1 \right) \frac{n_{r+1}}{n_r} \]

- a = 2 times \( \rightarrow r = 2 \)
- b = 1 time \( \rightarrow r = 1 \)
- c = 0 times \( \rightarrow r = 0 \)
- d = 1 time \( \rightarrow r = 1 \)
- e = 3 times \( \rightarrow r = 3 \)
- f = 3 times \( \rightarrow r = 3 \)
- g = 0 times \( \rightarrow r = 0 \)

\[ n_0 = 2 \]
\[ n_1 = 2 \]
\[ n_2 = 1 \]
\[ n_3 = 2 \]

\[ P'(c, g) = P'_0 = \frac{1}{10} \times \frac{2}{2} = \frac{1}{10} \]
\[ P'(b, d) = P'_1 = \frac{2}{10} \times \frac{1}{2} = \frac{1}{10} \]
\[ P'(a) = P'_2 = \frac{3}{10} \times \frac{2}{1} = \frac{6}{10} \]
\[ P'(e, f) = P'_3 = \frac{4}{10} \times \frac{3}{2} = \frac{6}{10} \]

Use interpolation for higher counts or lacks