Prioritizing Alerts from Static Analysis with Classification Models

Problem

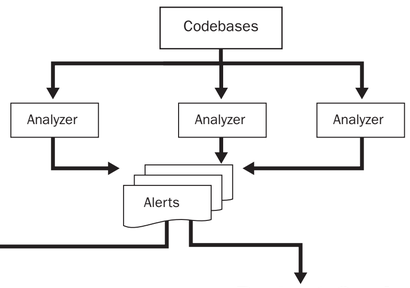
The number of security-related code flaws detected by static analysis requires too much effort to triage.

Significance

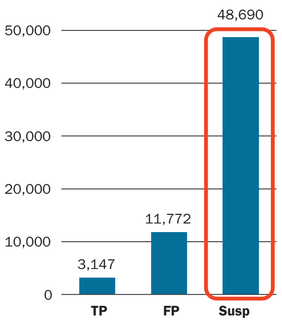
- Code flaws and vulnerabilities remain
- Scarce resources are used inefficiently

Project goals

Classification algorithm development using CERT- and collaborator-audited data, to accurately estimate the probability of true & false positives, intended to reduce analyst effort.

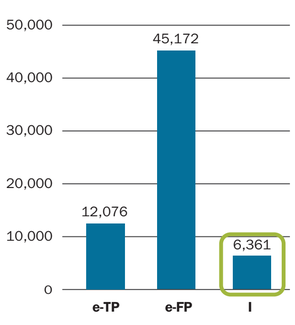


Today



Many alerts left unaudited! (red box)

Project Goal



Prioritized, small number of alerts for manual audit (green box)
Most alerts automatically "audited" by classifier as expected True (e-TP) or False (e-FP)

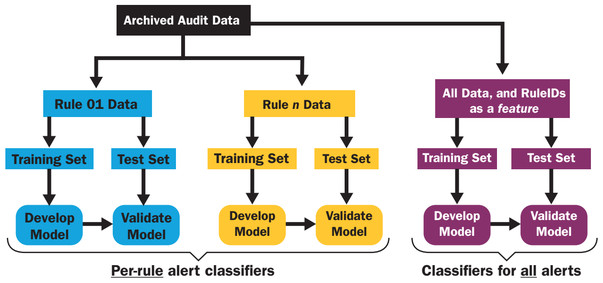
Classification algorithm development using CERT-and collaborator-audited data, that **accurately classifies most of the diagnostics as:** Expected True Positive (e-TP) or Expected False Positive (e-FP), and the rest as Indeterminate (I)

Scientific Approach

Novel combined use of:

1) multiple analyzers, 2) variety of features,

3) competing classification techniques!



Competing Classifiers to Test
Lasso Logistic Regression
CART (Classification and Regression Trees)
Random Forest
Extreme Gradient Boosting (XGBoost)

Some of the features used (many more)
Analysis tools used
Significant LOC
Complexity
Coupling
Cohesion
SEI coding rule

Data Used for Classifiers

Data used to create and validate classifiers:

- CERT-audited alerts:
  - ~7,500 audited alerts
- 3 DoD collaborators audit their own codebases with enhanced-SCALe

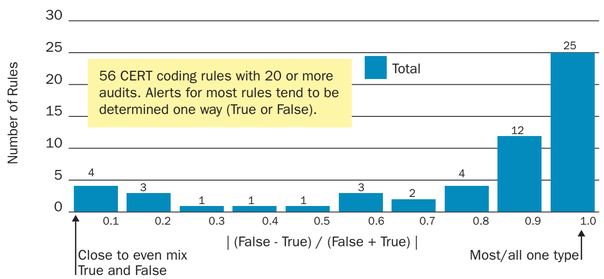
We pooled data (CERT + collaborators) and segmented it:

- Segment 1 (70% of data): train model
- Segment 2 (30% of data): testing

Added classifier variations on dataset:

- Per-rule
- Per-language
- With/without tools
- Others

CERT-audited data



Classifier Test Highlights

Classifiers made from all data, pooled:

All-rules (**158 rules**) classifier accuracy:

- Lasso Logistic Regression: 88%
- Random Forest: 91%
- CART: 89%
- XGBoost: 91%

Single-rule classifier accuracy:

Rule ID	Lasso LR	Random Forest	CART	XGBoost
INT31-C	98%	97%	98%	97%
EXP01-J	74%	74%	81%	74%
OBJ03-J	73%	86%	86%	83%
FIO04-J	80%	80%	90%	80%
EXP33-C*	83%	87%	83%	83%
EXP34-C*	67%	72%	79%	72%
EXP36-C*	100%	100%	100%	100%
ERR08-J*	99%	100%	100%	100%
IDS00-J*	96%	96%	96%	96%
ERR01-J*	100%	100%	100%	100%
ERR09-J*	100%	88%	88%	88%

*Single-rule IDs with asterisk: small quantity of data, results suspect

General results (not true for every test)

- Classifier accuracy rankings for all-pooled test data: XGBoost ≈ RF > CART ≈ LR
- Classifier accuracy rankings for collaborator test data: LR ≈ RF > XGBoost > CART
- Per-rule classifiers generally not useful (lack data), but 3 rules are exceptions.
- With-tools-as-feature classifiers better than without.
- Accuracy of single language vs. all-languages data: C > all-combined > Java

288 Classifiers Developed

- 15 featureless classifiers (20 or more audits, 100% True or False)
- 201 classifiers for 11 with mixed determinations
  - True/False ratio & count combination insufficient for classifiers, for some rules
- 72 all-rules classifiers name used as feature
  - 44 per-language classifiers

Results with DoD Transition Value

Software and paper: Classifier-development

- Code for developing classifiers in R
- Paper on classifier project [1]

Software: Enhanced-SCALe Tool (multi-tool alert auditing framework)

- Added data collection
- Archive sanitizer
- Alert fusion
- Offline SCALe installs and first VM

Training to ensure high-quality data

- SEI CERT coding rules
- Auditing rules [2]
- Enhanced-SCALe use

Auditor quality test

- Test audit skill: mentor-expert designation

Conference/workshop papers from project:

[1] Flynn, Snavely, Svoboda, Qin, Burns, VanHoudnos, Zubrow, Stoddard, and Marce-Santurio. "Prioritizing Alerts from Multiple Static Analysis Tools, using Classification Models", work in progress.

[2] Svoboda, Flynn, and Snavely. "Static Analysis Alert Audits: Lexicon & Rules", IEEE Cybersecurity Development (SecDev), November 2016.

Future work

Goal: improve accuracy

- Try different classification techniques
- Add features:
  - Semantic features (ICSE 2016)
  - Dynamic analysis tool results
- More audit archive data needed
  - Additional data welcome! Potential collaborators, please contact me
  - FY17 project focuses on rapid expansion of per-rule classifiers

Distribution Statement A: Approved for Public Release; Distribution is Unlimited

Contact: Lori Flynn, PhD | lflynn@sei.cmu.edu