ALETHEIA: Improving the Usability of Static Security Analysis

JAN 13TH, 2016

论文下载：http://researcher.watson.ibm.com/researcher/files/us-otripp/ccs14.pdf

Abstract

- 静态分析efficient and scalable
- 但是存在误报，影响了可用性
- 提出改进静态分析结果的一般性方法：根据用户的决策“有监督”机器学习
- （工业界的文章，方法新意不大，同时把决策的责任都给了用户，但可以快速实现）

Introduction

- 静态安全分析并不是完美的：为了大规模化，就必须放松标准、采取近似的策略
- 精度损失的方面：flow insensitivity, path insensitivity, context insensitivity
- 用户对少数原生问题分类；确定祛除false negtive和保护true positive的策略
- （策略其实是在precision和recall间权衡）

Overview

Limitations of Static Analysis

- Flow insensitivity

```
x.f = read(); x.f = ""; write(x.f);
```

1 分析不会跟踪内存更新顺序
2 以上不会记录x.f的更新，只会记录被赋值了不可信数值
3 Path insensitivity

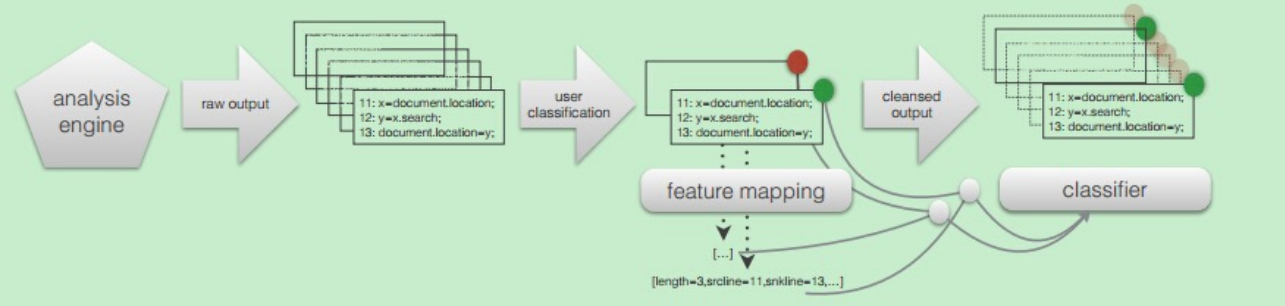
```
x.f = ""; if (b) { x.f = read(); } if (!b) { write(x.f); }
```

1 流问题
2 会分析不可达路径
3 Context insensitivity

```
y1 = id(x); y2 = id(read()); write(y1);
```

1 id()是类似echo的返回输入的函数
2 第一次调用的时候是可信的，但第二次调用时是不可信的，综合起来就是id()可能是不可信的

System Achitecture



Learning Features

- 选取的features会导致误报
- （针对js）

Lexical Features

- source/sink identifier：field, function的名字，如document.location
- sink line number
- source/sink URL：包含source/sink语句的JS函数的URL
- external objects
- 语法信息对发现第三方库、组件使用是有效果的

Quantitative Features

- total results on: the overall number of fuindings reported on the file containing the sink statement
- number of steps: the number of flow milestones comprising the witness path
- time
- number of path conditions
- number of functions

Security-specific Features

- 用户定义

Learning Algorithms

- 大概介绍几种分析方法
- （介绍比较概括，启发性不是很强）

Functional Method

- 包括logistic regression(逻辑回归)，linear support vector machines and generalizations, such as neural nets(神经网络)
- 线性方法有一个问题：the richness of the model space – there are limits to how well a linear classifier can perform（模型空间太丰富，线性分类器有性能上限）

Instance-based Classification

- 用distance function计算实例间的距离，如Kstar算法

Tree- and Rule-based Methods

- 分治方法根据标签(labels)快速分开数据实例，如决策树
- 基于规则，顾名思义，就是规定分类的规则

Bayesian Methods

$$P(C = c \mid X = x) = \frac{P(X = x \mid C = c)P(C = c)}{P(X = x)},$$

Discussion

- 对于静态安全分析，“几何”方法，如向量机、逻辑回归模型，还有基于欧几里得距离的Instance-based方法都是不合适的
- 因为提取有价值的数值信息是有难度的
- 所以非“几何”方法，如基于树、规则的方法是比较好的选择
- （1R算法）

Implementation and Evaluation

Prototype Implementation

- 作为Java library实现
- 在Weka 3.6.10基础上实现：点我传送

$$p = \frac{tp}{tp + fp} \quad (precision) \qquad (2)$$

$$r = \frac{tp}{tp + fn} \quad (recall) \qquad (3)$$

- P：结果当中有多少是准确的；R：有多少准确的被找出来了
- 用以下公式调和P和R

$$w \times r + (1 - w) \times p \quad (4)$$

Experimental Setup

- 用现有的JS security checker(没有指明工具)分析了来自675个最热门网站的1760个HTML网页，得到3758个warning(多向性表明)
- 做如下实验：
  1. 从3758个warnings中随机抽取出n个
  2. 平分成2份，一份做训练，一份做测试，用可用的所有分类器
  3. 分类结果应用于剩下所有warnings，计算P和R

Experimental Results

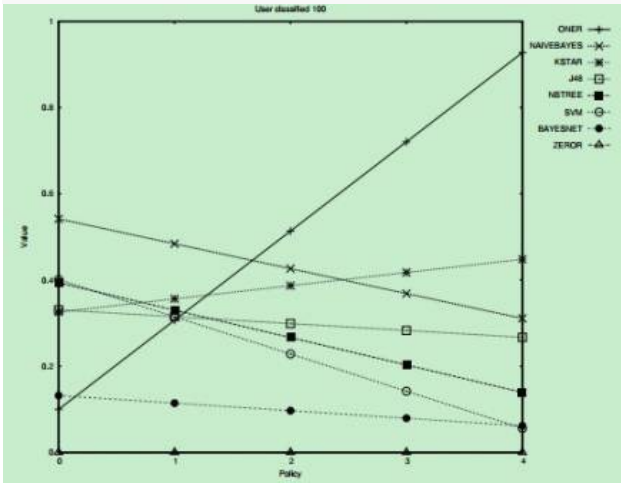


Figure 3: Scores Achieved by the Different Classifiers As a Function of the Policy Given 100 Classified Warnings

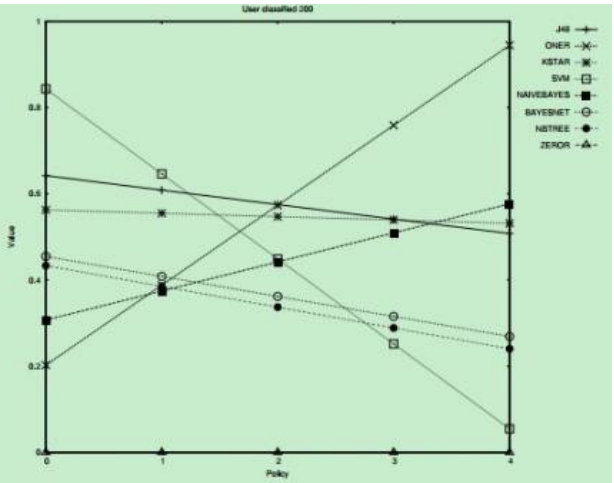


Figure 4: Scores Achieved by the Different Classifiers As a Function of the Policy Given 200 Classified Warnings

- policy代表权重值w的分子i, w == i / 4

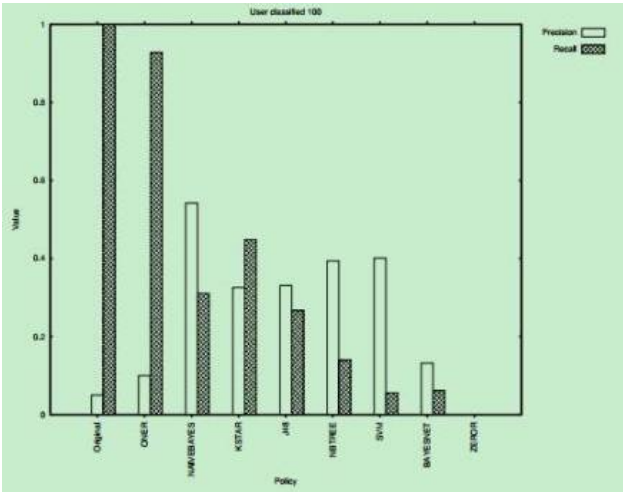


Figure 5: Precision and Recall for the Different Classifiers Given 100 Classified Warnings

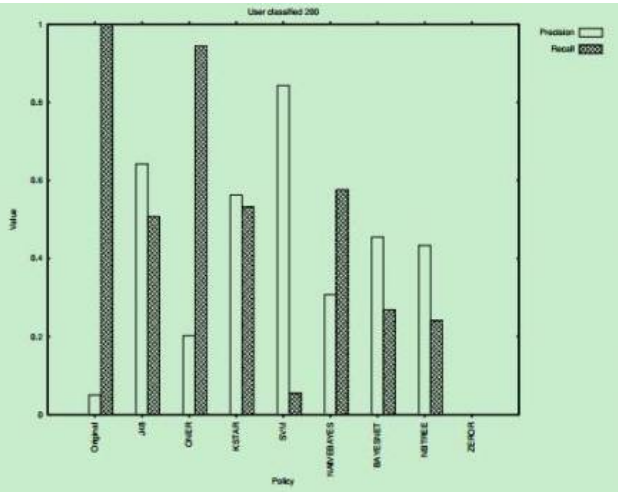


Figure 6: Precision and Recall for the Different Classifiers Given 200 Classified Warnings

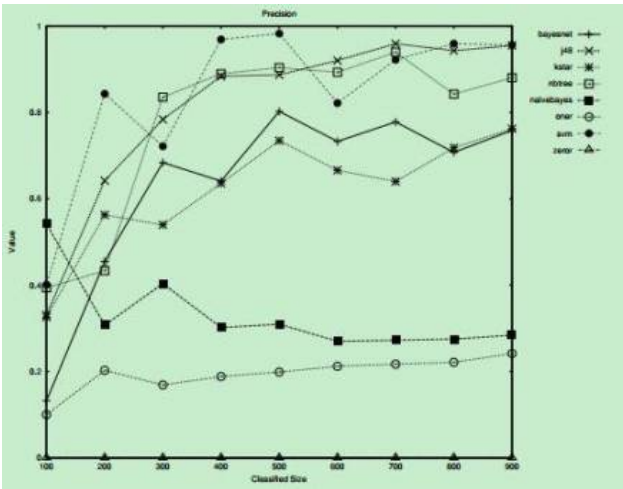


Figure 7: Precision As a Function of Classified-set Size

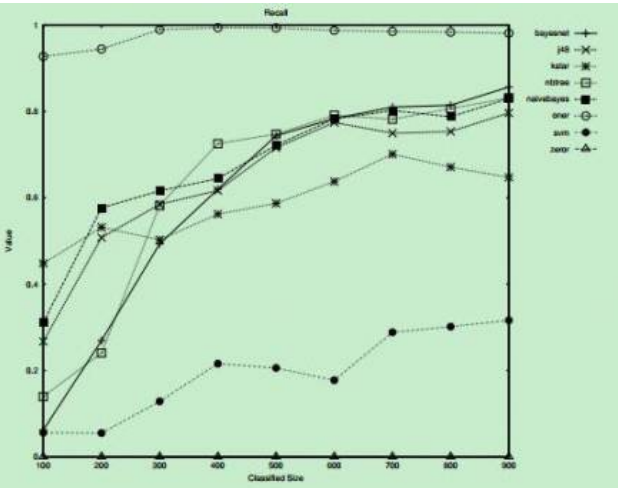


Figure 8: Recall As a Function of Classified-set Size

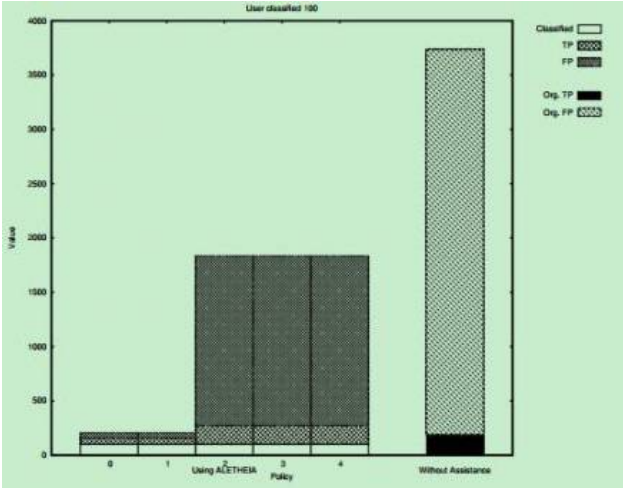


Figure 9: Number of Findings the User Has to Review with ALETHEIA (by Policy: 1-4) and without ALETHEIA Given 100 Initial Classifications

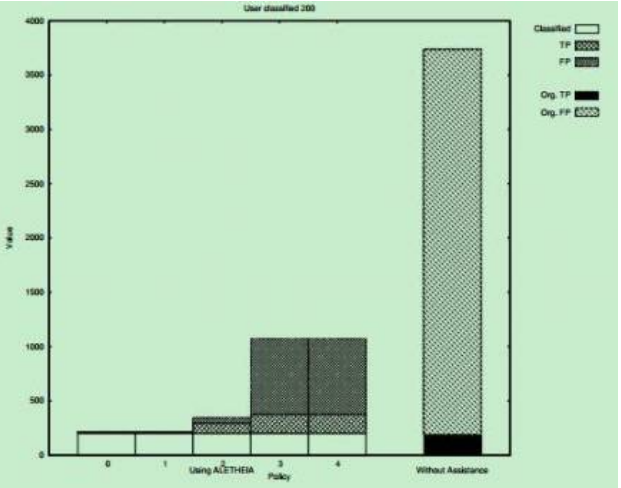


Figure 10: Number of Findings the User Has to Review with ALETHEIA (by Policy: 1-4) and without ALETHEIA Given 200 Initial Classifications

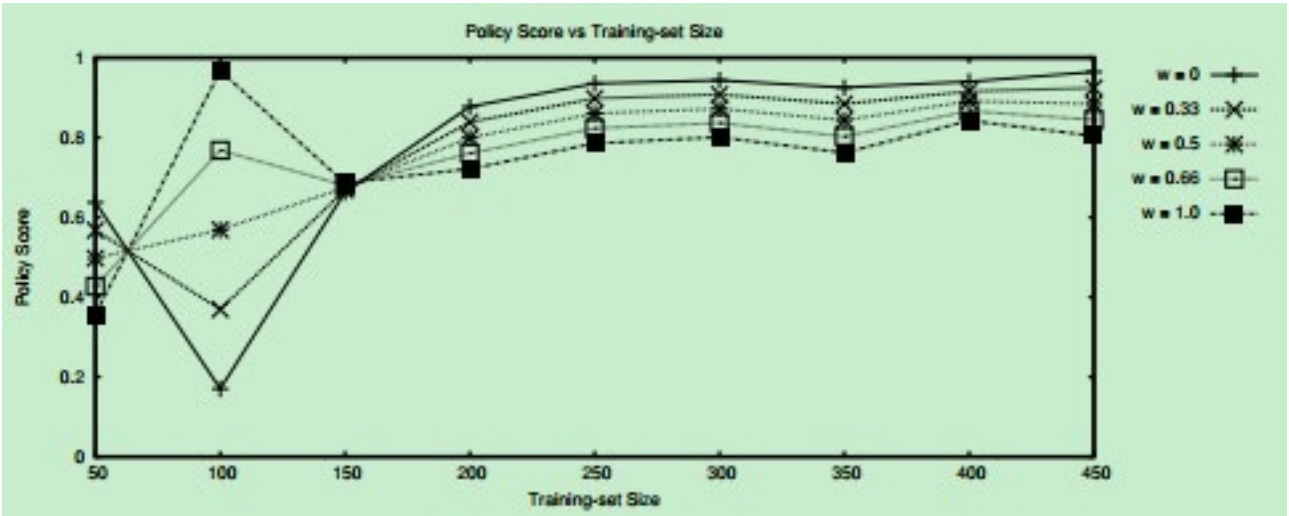


Figure 13: Policy Score as a Function of the Training-set Size, where Policies Are Represented as Their Respective w Value