ALETHEIA: Improving the Usability of Static Security Analysis

ALETHEIA- Improving the Usability of Static Se.pdf
2018-08-11 10:50:13, 1.19 MB

术语

- information-flow vulnerability
- source：读取不受信任的用户输入（或获取敏感信息，如用户位置）的语句
- sink：执行安全相关操作，如更新数据库（或发布信息）的语句
- downgrader：对输入进行验证（或解密敏感数据）的语句
- witnesses（counterexamples）：source和sink之间的一条downgrade-free path

---

Abstract

- 软件规模和复杂度提高→人工安全审计复杂
- 自动化静态分析高效，但是 `误报率高` (可用性低)
- 提出改进静态分析结果的一般性方法：基于用户决策的“有监督”机器学习（基于用户对部分warning的警告反馈的分析，将机器学习方法应用到报告输出）
- 将决策的责任甩给了用户

Request for data in A LETHEIA : Improving the Usability of Static Security Analysis of Static Security Analysis

```
Dear author,

    I am a postegraduate student in Nanjing University, China.

    I am studying your awesome paper-A LETHEIA : Improving the Usability of Static Security Analysis. I want to recurrent your work, but I cannot get the data(1,700 HTML pages) in the experiment.

    Could you send me the data?
```

```

Best wishes!
```

Introduction

1. 静态分析作用
   1.1. 对于分析information-flow vulnerability（完整性（XSS、XAS）、机密性（敏感数据泄露））有效
   1.2. Static Information-flow Analysis（污点分析，解决可达性问题（sources和sinks之间的可达性））
2. 静态分析缺陷
   2.1. 为了大规模化，必须采用近似的策略→`误报`
   2.2. 精度损失：flow insensitivity, path insensitivity, context insensitivity
3. 提高静态分析有效性的方法
   3.1. 方法的基本要求：普遍性（只针对warnings，不涉及检测工具）、可定制化（用户可以自己权衡precision和recall）
   3.2. 实际操作：

- 用户：对部分原生数据（报告）分类；确定去除false positive和保留true positive之间的权衡

Overview

1.1. 静态分析的局限性-分析程序运行时行为固有的局限、为了大规模化而牺牲精度
1.2. 支持大规模的设计

- 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

- 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

- 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. 第一个调用的时候是可信的，但第二个调用时是不可信的，综合起来就是id()可能是不可信的

System Architecture

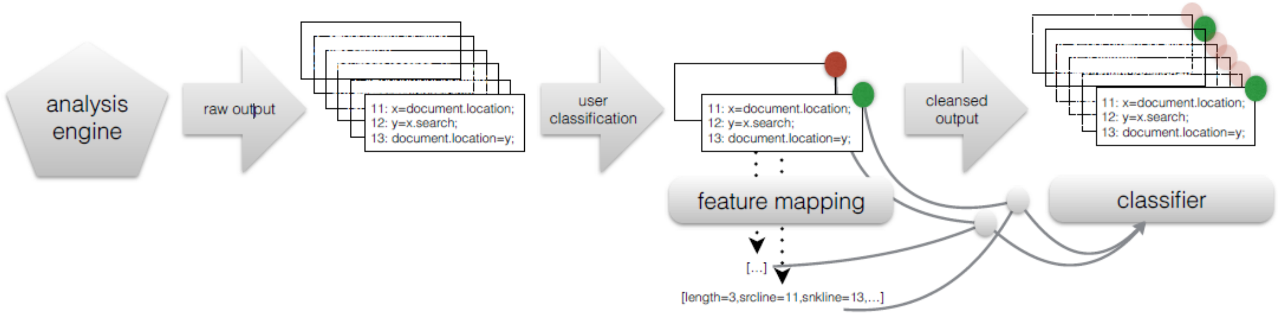

Figure 2: Visual description of the workflow of the ALETHEIA system

---

Learning Features

Lexical Features

- source/sink id：source/sink语句的field, function的名字，如document.location
- source/sink line number：行号
- source/sink URL：包含source/sink语句的JS函数的URL

- external objects：执行嵌入功能（如Flas）的flag
- 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 (results): The overall number of findings reported on the file containing the sink statement.
- Number of steps (steps): The number of flow milestones comprising the witness path.
- Time (time): The total time spent by the analysis on the scope containing the witness.
- Number of path conditions (conditions): The number of branching statements (either loops or conditions) along the witness path.
- Number of functions (functions): The number of functions enclosing statements along the witness path.

Security-specific Features

- rule name
- severity

Learning Algorithms

- 大概介绍以下四种方法
- 介绍比较概括，启发性不是很强

Functional Methods

- 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

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

Implementation and Evaluation

Prototype Implementation

- 作为Java library实现
- 在Weka 3.6.10基础上实现
- p(precision，精确率，结果当中有多少是准确的)和r(recall，召回率，有多少准确的被找出来了)

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

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

- 在precision和recall之间权衡，w∈{0/4, 1/4, 2/4, 3/4, 4/4}

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

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(多样性表明)
- 实验步骤如下：
  a. 从3758个warnings中随机抽取出n个
  b. 将n平分成2份，一份做训练，一份做测试，用可用的所有分类器
  c. 分类结果应用于剩下所有warnings，计算P和R

Experimental Results

- Policy: w∈{0/4, 1/4, 2/4, 3/4, 4/4}

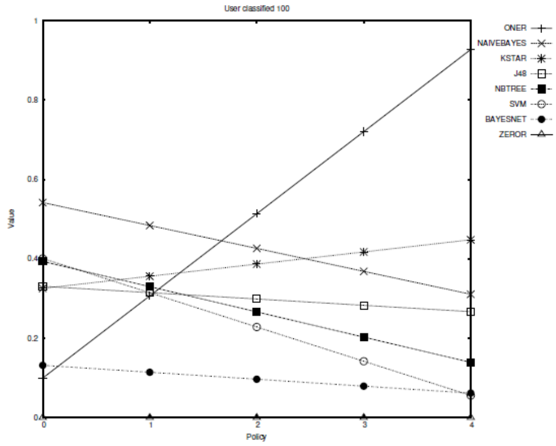

Figure 3: Scores Achieved by the Different Classifiers As a Function of the Policy Given 100 Classified Warnings

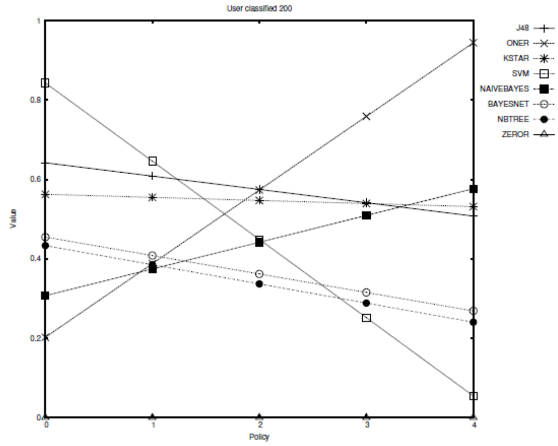

Figure 4: Scores Achieved by the Different Classifiers As a Function of the Policy Given 200 Classified Warnings

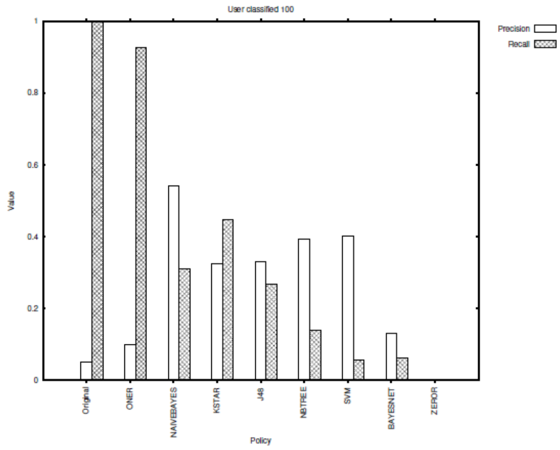

Figure 5: Precision and Recall for the Different Classifiers Given 100 Classified Warnings

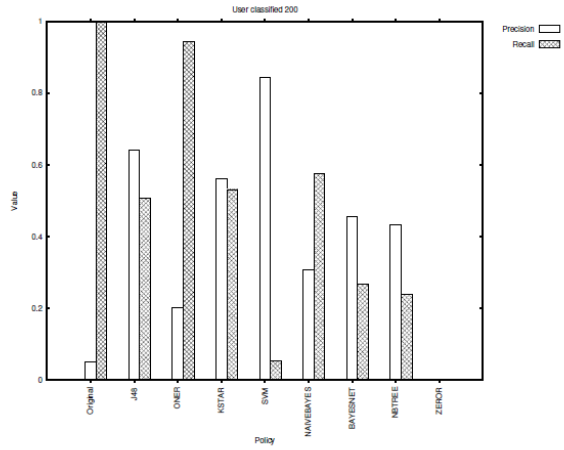

Figure 6: Precision and Recall for the Different Classifiers Given 200 Classified Warnings

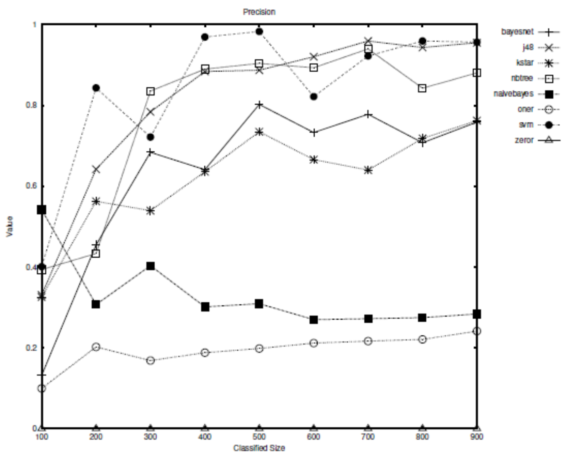

Figure 7: Precision As a Function of Classified-set Size

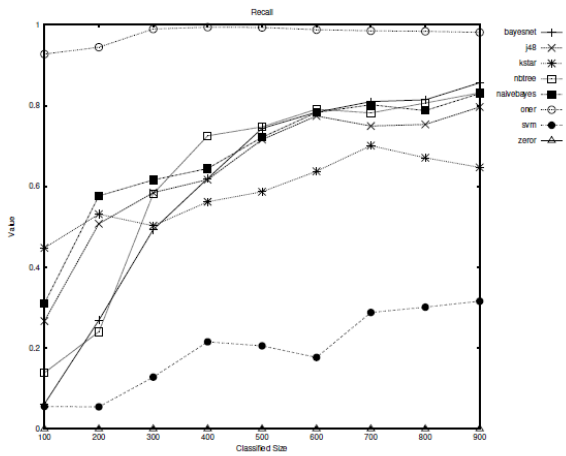

Figure 8: Recall As a Function of Classified-set Size

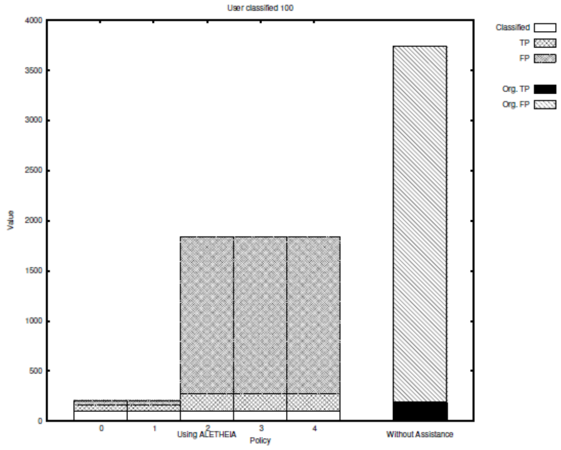

Figure 9: Number of Findings the User Has to Review with ALETHEIA (by Policy: 1-4) and without ALETHEIA Given 100 Initial Classifications

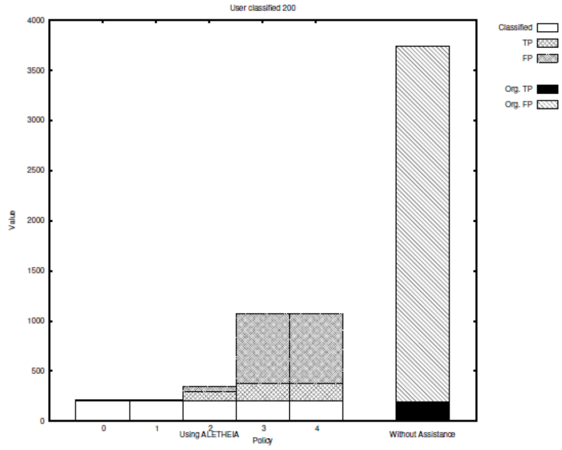

Figure 10: Number of Findings the User Has to Review with ALETHEIA (by Policy: 1-4) and without ALETHEIA Given 200 Initial Classifications

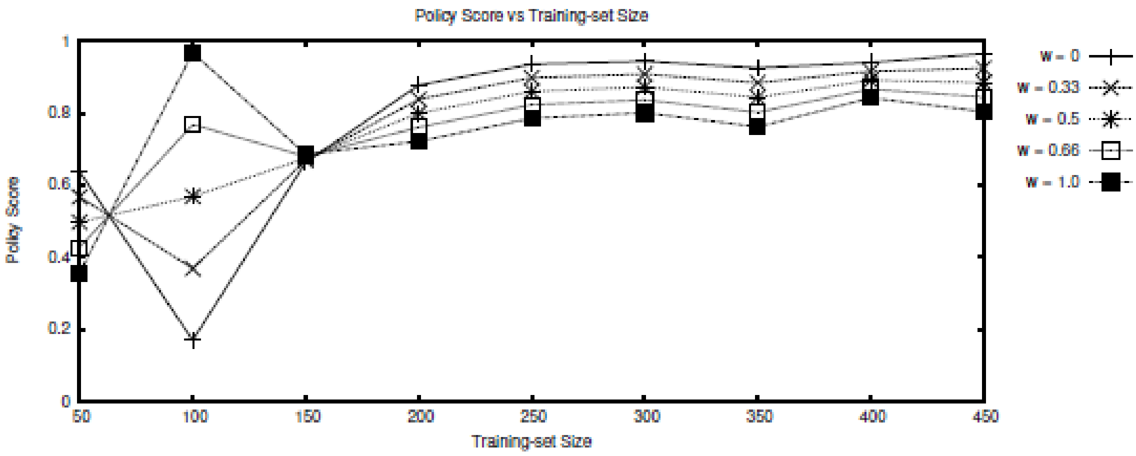

Figure 13: Policy Score as a Function of the Training-set Size, where Policies Are Represented as Their Respective w Value

---

问题：

1. 数据，false warnings比true alarms多很多

- 静态分析工具的价值
- 静态分析工具的缺陷
- 静态分析工具缺陷产生的原因

疑问

机器学习方法如何解决近似带来的误报

是否有办法获取到source、sink等扫描器相关信息，如果扫描工具不提供相应的信息