# Deep X-Ray: 一种机器学习驱动的WAF规则窃取器

XunSu
腾讯朱雀实验室AI安全研究员

KeyunLuo
腾讯朱雀实验室AI安全研究员

# 团队介绍

XunSu

- 10年安全从业经验
- 专注于java安全和渗透测试
- 现从事AI辅助攻击方面的研究工作

KeyunLuo

- 4年安全AI研究经验
- 擅长自然语言处理、数据挖掘

腾讯安全平台部
Tencent Security Platform Dpt.

腾讯朱雀实验室
Tencent Zhuque Lab

**安全平台部朱雀实验室**

安全平台部下设的安全实验室，致力于开展实战级别的APT攻击和AI攻击技术研究，以攻促防，守护腾讯安全

# WAF规则逆向的意义

- 复制WAF的防护能力
- 更容易构造出绕过WAF的Payload

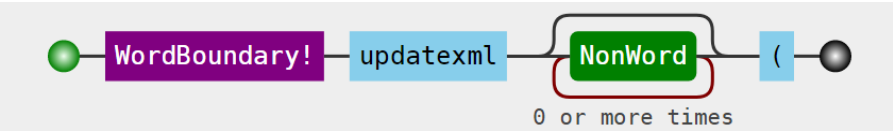
modsecurity
Open Source Web Application Firewall

## 提出问题

如何逆向出WAF规则？

# 逆向规则的原理和方法

(?i)\bupdatexml\W*\(




| Regex | Block | Pass |
|---|---|---|
| \w | 0-9a-zA-Z_ | !@#... |
| \W | !@#... | 0-9a-zA-Z_ |
| \d | 0-9 | a-zA-Z_!@#... |
| \D | a-zA-Z_!@#... | 0-9 |
| \s | [ \f\n\r\t\v] | 0-9a-zA-Z_!@#... |
| \S | 0-9a-zA-Z_!@#... | [ \f\n\r\t\v] |
| \b | 1aA_ | !@#$%^&*() |

## 逆向规则的原理和方法

- 找出最小匹配单元作为种子payload，例如updatexml(
- 构造正负样本payload测试规则得到反馈，推出正则单元
- 最后将推测出的正则单元合并得到WAF规则
- 另外构造一批正负样本，测试推测出的WAF规则是否严谨

## 步骤一：测试最小匹配单元（种子payload）

| 正反馈 | |
|---|---|
| updatexml( | block |
| updatexml() | block |
| (updatexml( | block |

+

| 负反馈 | |
|---|---|
| updatexml | pass |
| pdatexml( | pass |

推出临界点

**updatexml(**

# 步骤二：测试边界合并正则单元

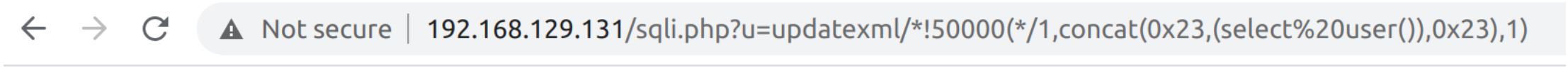
| | | | | |
|---|---|---|---|---|
| updatexml( block<br>UpdaTeXml( block | | | → | (?i) |
| 正反馈<br>updatexml( block<br>a updatexml( block<br>(updatexml( block | + | 负反馈<br>1updatexml( pass<br>aupdatexml( pass<br>_updatexml( pass | → | \b |
| 正反馈<br>updatexml/**/( block<br>updatexml!( block<br>updatexml@( block | + | 负反馈<br>updatexml1( pass<br>updatexmla( pass<br>updatexml_( pass<br>updatexml/*!123*/( pass | → | \W |

合并

(?i)\bupdatexml\W*\(

## 已知规则构造绕过payload

**(?i)\bupdatexml\W*\(** → **updatexml/*!50000(*/1,concat(0x23,(select user()),0x23),1)**



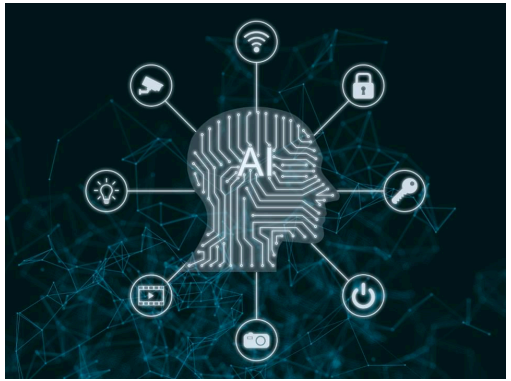
## 思考：可以自动化完成吗

### 人工探测的不足之处

- 依赖攻击者的技术水平
- 重复工作，耗时耗力难以大规模展开
- 细节方面很容易疏漏
- 人工经验难以覆盖全量的攻击payload

### 机器学习的优势

- 有限的人工介入
- 善于从数据中学习规律
- 可以批量的自动化运行



## 解决方案

数据 + 算法 + 探测 → 自动化输出规则

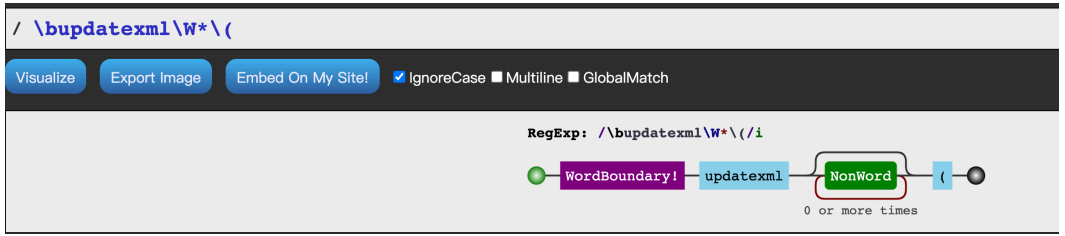
## 数据


- \\bexecute\\(
- order\\sby
- (?i)\\d\\s+group\\s+by.+\\(
- ……

遍历规则

1 group by abc(
a1 group by abc(b
1groupby abc(
……

# 数据




re.sre_parse.parse(r'\bupdatexml\W*\(').dump()

```
AT AT_BOUNDARY
LITERAL 117
LITERAL 112
LITERAL 100
LITERAL 97
LITERAL 116
LITERAL 101
LITERAL 120
LITERAL 109
LITERAL 108
MAX_REPEAT 0 MAXREPEAT
  IN
    CATEGORY CATEGORY_NOT_WORD
LITERAL 40
```

| | |
|---|---|
| updatexml( | 1updatexml( |
| updatexml!( | updatexml2( |
| @updatexml( | _updatexml( |
| … | … |

# 问题1：如何学习payload中包含的安全经验?

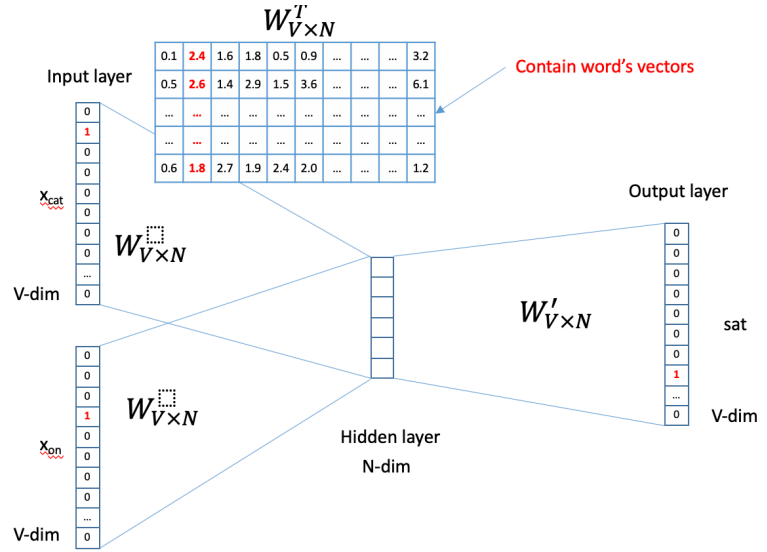
- 自动化提取Payload的文本特征
  - 搭配用法
  - 相似用法

# 算法：预训练模型

A quick brown fox jumps over the lazy dog

词向量模型

- 方式：无监督学习
- 任务：预测中心词/邻居词



# 问题2：如何自动化产生用于推测的种子payload?

- 控制变量，便于探测
  - 子序列抽取

updatexml (1,concat(0x23,(select user()),0x23),1)

↓

[updatexml , concat, select, user]

# 算法：注意力机制

The first recorded travels by Europeans to China and back date from this time. The most famous traveler of the period was the Venetian Marco Polo, whose account of his trip to "Cambaluc," the capital of the Great Khan, and of life there astounded the people of Europe. The account of his travels, Il milione (or, The Million, known in English as the Travels of Marco Polo), appeared about the year 1299. Some argue over the accuracy of Marco Polo's accounts due to the lack of mentioning the Great Wall of China, tea houses, which would have been a prominent sight since Europeans had yet to adopt a tea culture, as well the practice of foot binding by the women in capital of the Great Khan. Some suggest that Marco Polo acquired much of his knowledge through contact with Persian traders since many of the places he named were in Persian.

How did some suspect that Polo learned about China instead of by actually visiting it?

**Answer:** through contact with Persian traders

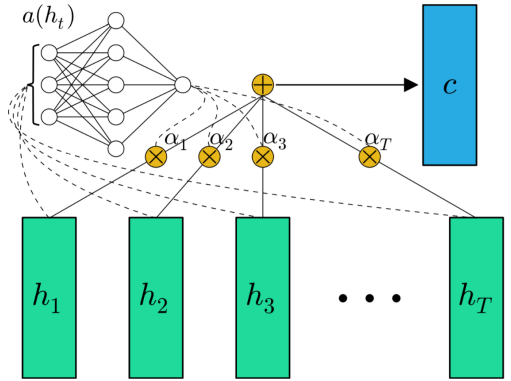
- 当一个人在吵闹的鸡尾酒会上和朋友聊天时，尽管周围噪音干扰很多，他还是可以听到朋友的谈话内容，而忽略其他人的声音。
- 同时，如果未注意到的背景声中有重要的词（比如他的名字），他会马上注意到。

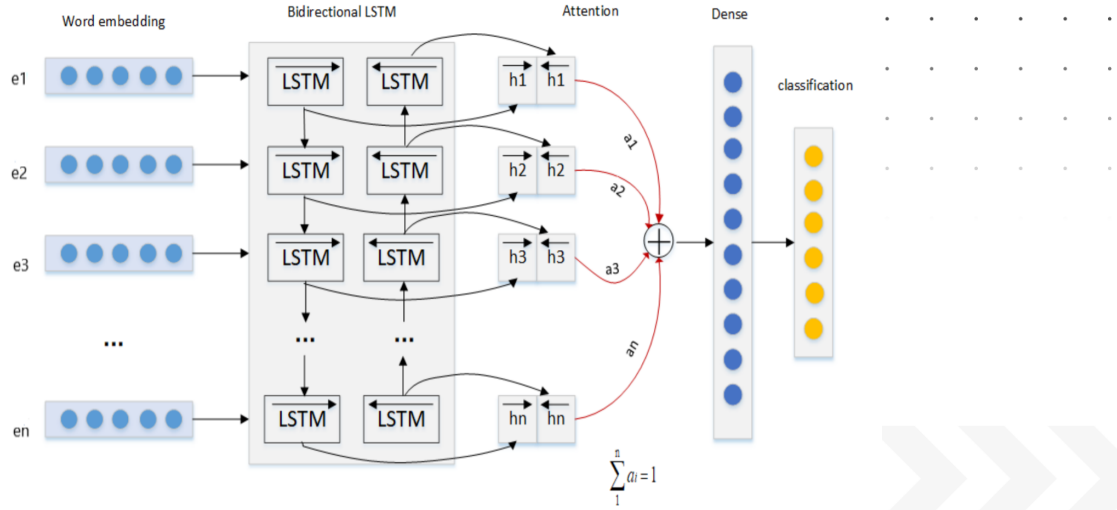
什么是注意力机制

# 算法：注意力机制

注意力机制：

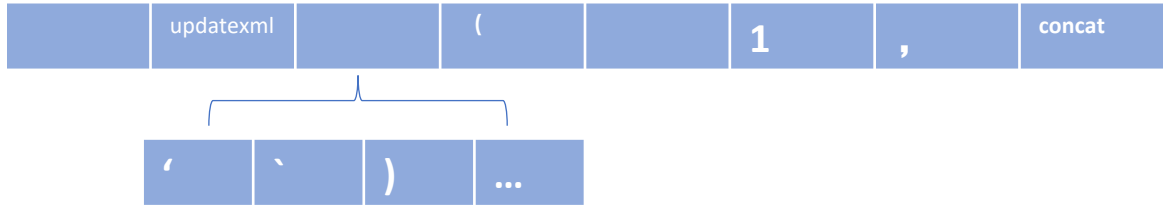
- 计算每个单词对结果的重要性
- 缩小搜索空间和尝试次数



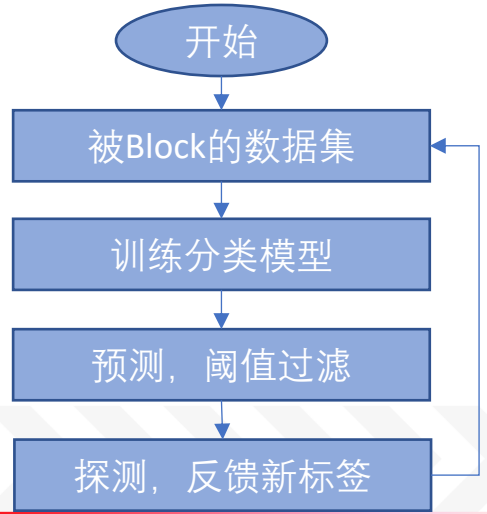



# 问题3：如何快速确定某个位置上该用哪个单词来探测?

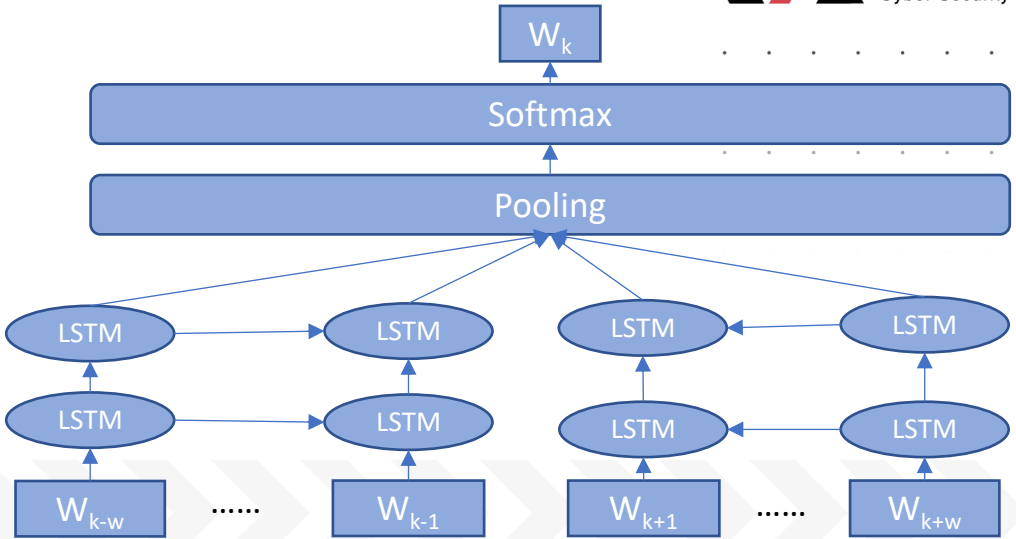
- 经验自动化
  - 抽取依据经验产生候选探测集
  - 根据探测结果修正预期
  - 记住该环境下的习惯性搭配
  - 应用在新payload中



## 算法：推荐

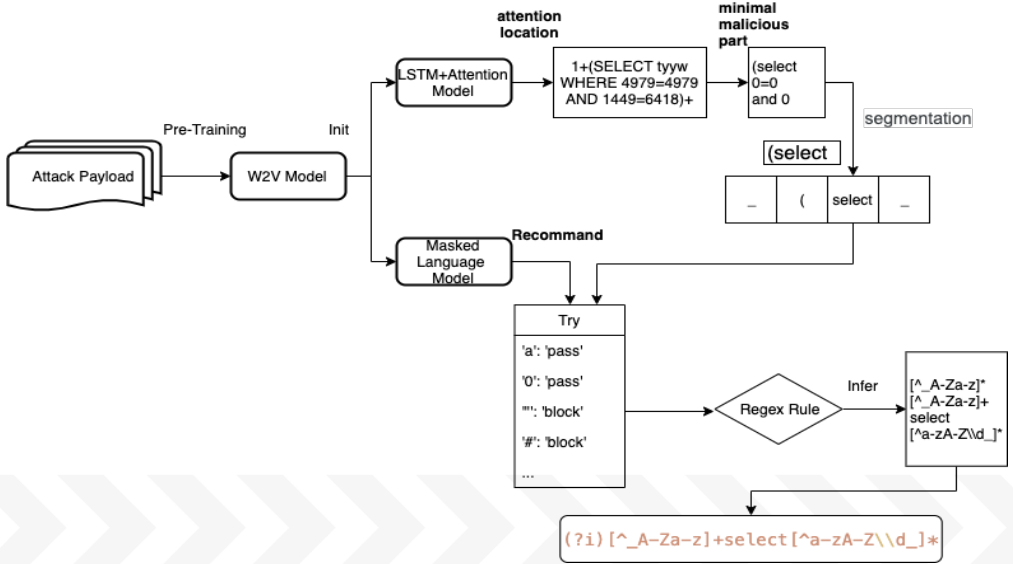



增量学习
持续优化



# 流程

- 预训练词向量
- 注意力机制
- 推荐模型
- 正负探测字符规约



## 实验

1. 词向量训练
2. 种子payload生成
3. 规则探测&合并
4. 结果分析

# 实验：词向量训练

- **分词**：按特殊符号分割

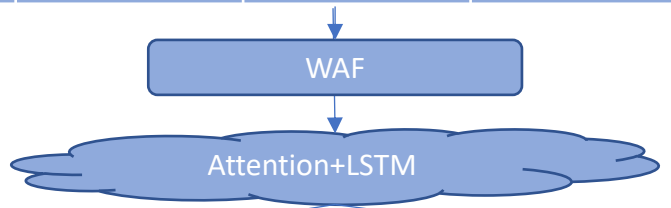
['~', '!', '@', '#', '$', '%', '^', '&', '*', '(', ')', '-', '+', '=', '{', '}', '[', ']', '|', '\\', ':', ';', '"', '\'', '<', '>', ',', '.', '?', '/']

- **训练参数**：
  - 窗口大小：10
  - 嵌入维度：32
  - 迭代次数：20
  - 向量模型：Skip Gram
  - 词表大小：780

```
tensor([[ 0.0928,  0.0444, -0.1118,  ..., -0.1361, -1.1211, -0.2490],
        [ 0.1147,  0.2807, -0.1121,  ..., -0.1055, -1.0731, -0.1153],
        [ 0.0818,  0.1254, -0.1286,  ..., -0.0392, -1.1212, -0.0036],
        ...,
        [-0.0152,  0.0212, -0.0953,  ...,  0.0097, -0.0410, -0.0028],
        [ 0.0431, -0.0240,  0.0473,  ..., -0.0124,  0.0761,  0.0456],
        [-0.0404, -0.0661,  0.0287,  ...,  0.0323, -0.0115, -0.0533]])
```

# 实验：种子payload生成

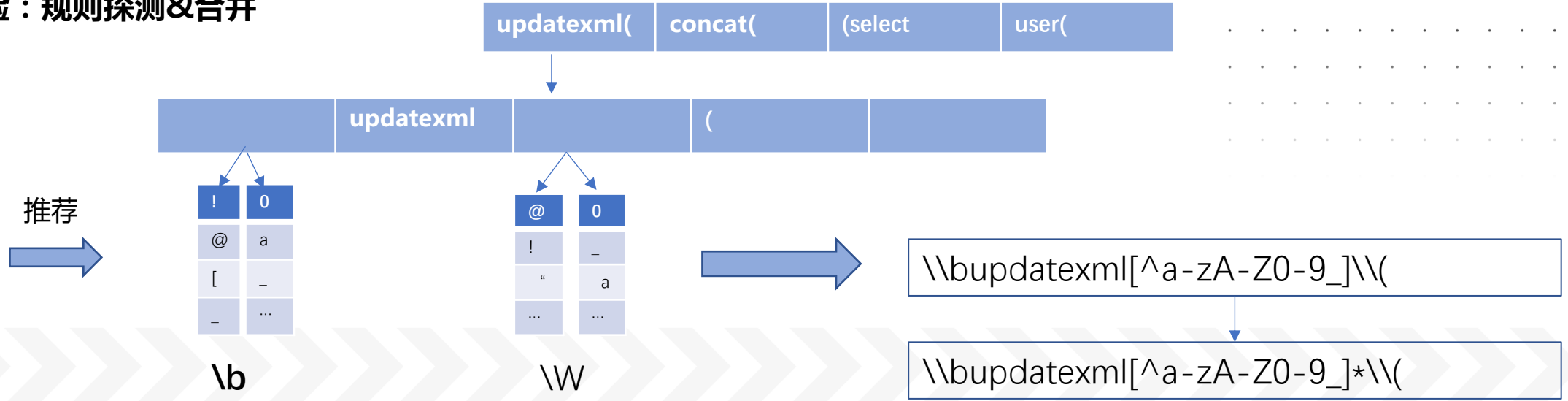
| updatexml | ( | 1 | , | concat | ( | 0x23 | , | ( | select |
|---|---|---|---|---|---|---|---|---|---|




| updatexml( | concat( | (select | user( |
|---|---|---|---|

种子　　种子

# 实验：规则探测&合并



# 实验：结果

```
\\bupdatexml[^a-zA-Z0-9_]*\\(
\\bconcat[^a-zA-Z0-9_]*\\(
[^a-zA-Z0-9_]*\\([^a-zA-Z0-9_]*select[^a-zA-Z0-9_]*
```

```
(?i)\\bupdatexml\\W*?\\(
(?i)\\bconcat\\W*?\\(
(?i)\\W*?\\(\\W*?select\\b
```

- 基于规则的检测模型：ModSecurity以及其他主流基于规则的WAF产品，拟合率在0.97以上
- 实现无专家干预下的一键规则窃取

# 场景启发

## 该方法可适用的类似场景

- 风控规则策略：根据线上反馈探测打击边界，实现绕过

## 该攻击难以覆盖的场景

- WAF使用基线模型分析请求并比较与正常业务的偏离程度
- 基于AI的后端策略，该策略很难以正则表达式的形式显式输出

## 启发

- 通过重写http响应以加深标签的不确定性，可以大大增加偷窃的难度
- 公开收集的payload中包含的安全经验，可以指导攻击者有效地执行自动化测试

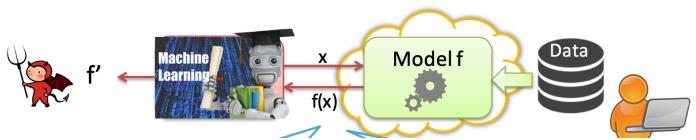
# 研究展望

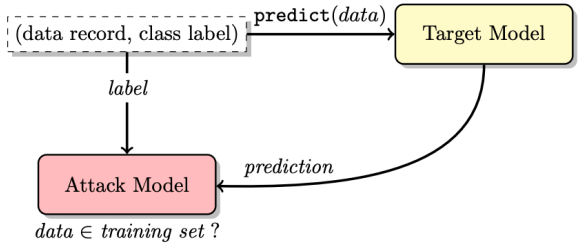
## 秒拨IP对抗

- 防止探测次数过多导致被封IP
- 并行处理，缩短窃取时间

## 语义模型 & AI模型能力克隆

- 构建本地影子模型
- 合成数据增强



# 研究展望

## 自动化Bypass实现

1. 应用文法将攻击payload分解成语法树
2. 语法树分解为子串
3. 初始训练集准备，对子串的排列组合形成的样本测试被WAF block的情况，获得原始标签
4. 利用监督学习决策树算法得到payload落在叶子结点上的决策路径
5. 机器学习驱动的进化算法测试策略：使用经典的基于种群的(μ + λ)EA进化算法生成后代，使用决策树(随机森林)分类结果作为适应度函数评判后代质量

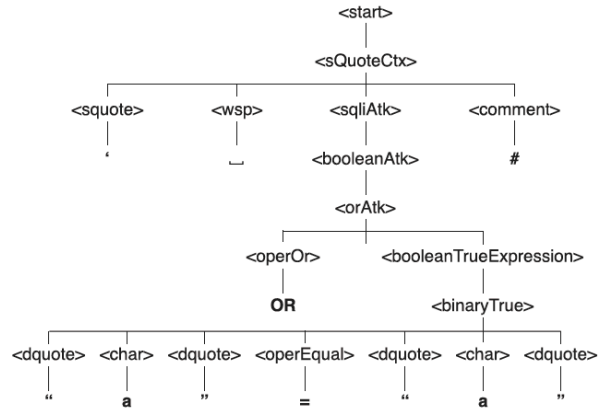
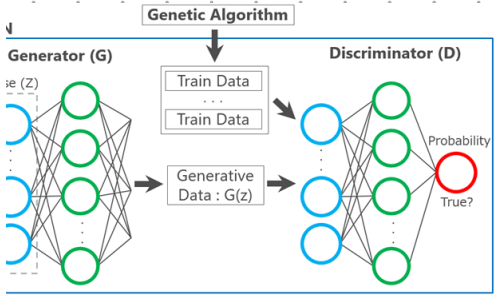



Fig. 2. Derivation tree of the “Boolean” SQLi attack: ′␣OR"a"="a"#.

基于进化算法的Bypass实现




基于GAN/Seq2Seq/EA/强化学习等模型的 Bypass payload自动生成

腾讯安全平台部
官方公众号

# THANKS

更多详情：https://deepxray.org/