# Web进程链异常指令检测算法实验环境验收报告

## 解释算法基本原理

在ppt中展示了算法原理，分析了检测方案的优缺点，指出了上线以后可能存在的问题以及可能的解决方案。

### 算法原理

n-gram距离可以用来衡量字符串之间的距离，我们使用归一化ngram距离作衡量指令之间的差异程度。训练时，我们随机选择的1000条恶意指令，无放回地取1000条正常指令作为测试数据，计算测试指令与所有正常样本的距离，取最小值作为该指令与白样本的距离，得到2000条测试数据与白样本的距离。我们设置0到1之间的距离阈值a，根据距离来判定指令正常与否，选择合适的阈值使模型准确率与召回率达到95%以上，作为模型的输出。

### 优缺点

优点：由于是无监督检测方案，所以具有对未知攻击的检测能力，当有一条之前尚未碰到过的指令时，判定为恶意指令。

缺点：检测比较耗时，因为要计算指令与正常数据集之间的距离。白样本安全性要有保障，混入恶意指令会导致检测准确率下降。由于是自定义算法，没有已有的工具包可用，需要自己手动实现，后期维护有难度。

### 上线后的问题

#### 误报

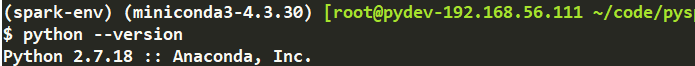
误报是需要考虑的情况，当发生误报的时候，我们将该指令加入白名单中，一段时间以后，我们使用白名单中的指令动态的训练模型

#### 告警抖动控制

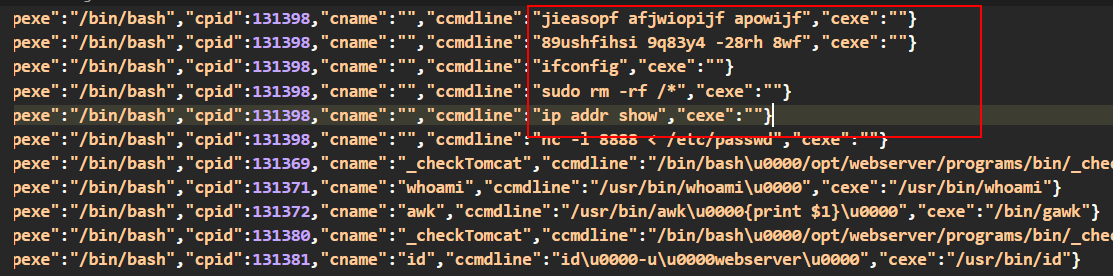
正常情况的告警曲线是较为平缓的，即使有攻击出现，执行的指令也是比较离散的，比如正常人输入执行的速度不可能达到5秒20条。当出现连续的，数量较多的恶意告警，我们倾向于认为线上产生了新业务。模型记录该业务的来源ip，来自该ip地址的指令将不再参与恶意命令检测。直到模型重新训练启动，该来源的指令都不再参与检测。

## 演示算法运行过程

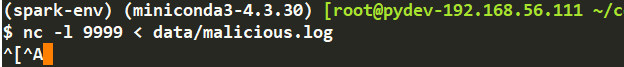
激活python虚拟环境（pyspark2.3.2 python2.7）



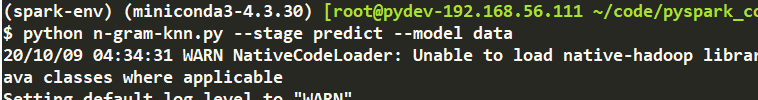
准备待测指令，格式与原始的训练日志格式相同



使用nc监听9999端口，并将待测指令重定向为标准输入



启动预测程序，stage参数标志当前执行预测过程，model参数表示模型的保存位置



程序检测出恶意指令，并且可以检测出未知指令

