创意书

一、选题说明

1.1、设计背景

1.1.1 项目名称

金科云盾

1.1.2 创意来源

大数据时代，信息变得唾手可得，庞大的中国互联网正遭受层出不穷的安全威胁，黑客的肆意攻击，让网络完全攻防站必须时刻戒备。众多公司和企业越来越重视通过网络手段辅助自己的企业运行，如通过微信小程序，支付宝小程序，后台人员信息管理网站等。来传递公司理念、商品、价值观，管理公司内部信息。与此同时，网络安全就变的尤为重要。虽然目前市场上出现了很多网络安全产品，但是大部分都是针对大型企业，跨国企业，国有公司等拥有庞大资金的部门。而与此相反的小型企业，小型部门，甚至个人博客主，受限于资金，往往没有选择安装安全软件。古语云：君子以思患而豫防之。所以市场就急需一款价格便宜，性能良好的安全软件，为它们的网站保驾护航。

我们的目标就是瞄准小型企业，众多小程序主，众多微商城主，甚至个人博客主，减轻经济负担的同时，又有出色的防护效果。

经过我们的考虑之后，我们的自命题在这样的背景下产生：网站遭受恶意攻击，通过 paddlepaddle 技术识别恶意攻击，阻止攻击的再次发生，保护网站安全。

1.2、市场调研

在了解我们产品之前，先了解一下目前市场上主流的高性能企业级的防火墙定价。

服务名称	服务价格	品牌
DDoS 高防 IP	21800.00/月	阿里
Web 应用防火墙	3880.00/月	阿里
Web 应用防火墙	3880.00/月	华为
网站管家 WAF	480.00/月	腾讯
Cloud Armor	$5/月+$1/规则 +$0.75/HTTP(S)请求	谷歌

AWS WAF	$5/月+$1/规则 +$0.60/百万 HTTP(S)请求	亚马逊

表 1 防火墙价格表

企业级防火墙的价格在几百到上万不等，这价格并不是一般的小型企业，营销号，微商城，甚至个人博主可以承担的。而且，这些企业级防火墙的功能，并不一定是这些小型企业所需要的。虽然小型企业未必有什么重要的数据信息，让黑客垂涎。但是同时也要考虑到，互联网是相互联系的，在这张网上，病毒的传播，攻击的发生，都是互相关联的。所以拥有安全软件是必要的，对比以上服务的售价，价格低廉的安全软件则变得很有市场。

于是，我们这样的产品就有了存在的可能性。

二、作品内容（paddlepaddle 应用）

2.1、作品（金科云盾-基于深度学习的网络云盾）流程图

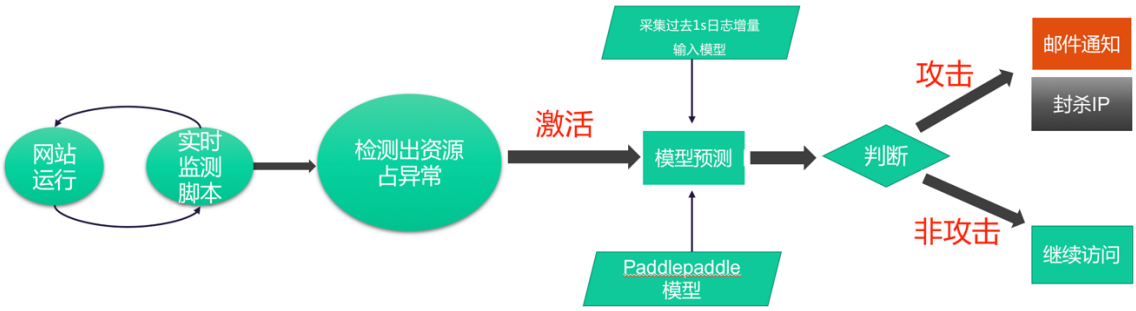


图 1 流程图（1）

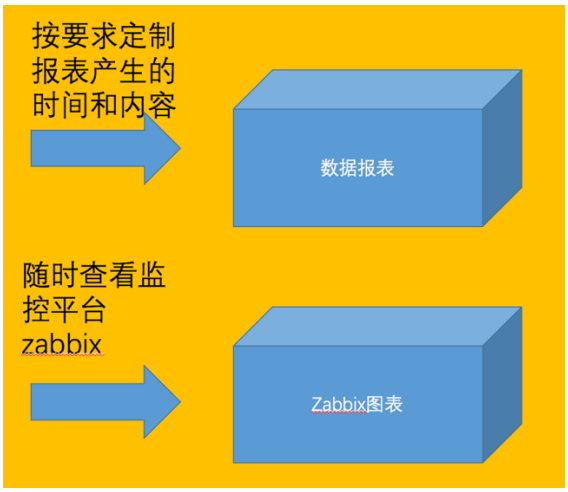


图 2 流程图（2）

通过数据处理手段，在监督学习的模式下，训练好 paddlepaddle 模型。将

模型部署到服务器上，设置好必要的选项，金科云盾就可以为您保驾护航了。

我们采用 apache 服务器，并设置日志格式为 NCSA。

当 apache 正常运行时，会产生指定格式的日志，通过自制脚本，自动检查日志文件的增量，当增量达到预先设置好的量，激活 paddlepaddle 模型。对日志增量进行预测，对预测结果自动分类，生成恶意访问报表，和正常访问报表。当识别到恶意访问，对其封杀 ip，限制其再次访问，保障网站安全。

定期自动整理报表数据，自动进行 paddlepaddle 训练，用来适应不同用户的不同网络环境，使得模型更加合理有效。报表数据也会以人性化的方式发送到客户邮箱上。

用户还可以定制可视化监控工具 zabbix，对于那些自建服务器的用户极大方便了服务器的管理。

2.2、技术特点

（1）小型化：磁盘占用量小，网络占用率低，内存占用率低，CPU 消耗低。

（2）快速部署：在辅助脚本的帮助下，可以实现一键环境测试，一键安装功能。

（3）兼容性好：与主流服务器 apache 和 nginx 有较好的兼容性。

（4）性能良好：针对主流攻击技术，做了针对性的防护，并且在测试中，效果良好。

（5）个性化处理方式：客户可以使用经过性能调优的监控软件 zabbix 实时观察服务器的动态，预测结果会定期制作成数据表报，对于大量密集攻击，也会有贴心的邮件告警机制。

（6）价格便宜：因为目标用户主要为大量的小型企业主、小型部门、微商城主、个人站长，所以采取低定价的策略。

三、作品功能实现

3.1、神经网络部分（paddlepaddle）

3.1.1 数据集的收集

为了制作真实的数据集，使用 django 结合 mysql 数据库和 apache2 服务器，apache 采用 NCSA 扩展/组合日志格式："%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\""。在百度云 BCC 服务器上搭建网站，网站地址为 106. 12. 38. 12。该页面实现了表单的填写，完成了与数据库的简单交互。且表单的内容不接受校验，也就说用户可以提交空表单。当多台机器，同时使用脚本模拟登录，并发送表单时，可以模拟出大量用户密集登录，这样就产生了正常的登录日志 common.log。为了能实现对恶意访问的识别，就要使用攻击工具。这里我们使用自制的 ddos 脚本，还有在 github 很火的 hping，慢链接工具采用 slowhttptest。采用这两种常见的攻击方式攻击网站。通过实验，这三个方法，都能使得，网站处于瘫痪状态，必须重启才能解决问题。证明攻击是有效的。在此情况下，收集日志 ddos.log 和 slow.log 作为我们的数据集。

3.1.2 数据集的制作

直接收集到的日志文件采用 NCSA 格式，有 9 个维度：远程主机 ip、远程登录名、远程用户名、请求的第一行、请求的状态、传输的字节数、客户端所用的浏览器版本信息、客户端的 HTTP 报头(host header)信息、客户端所用的协议等。文件内容用字符串存储，用空格分开。这样的数据是无法直接喂入神经网络的，必须对数据进行清洗。清洗的第一步是把字符类型数据转化成数值类型数据。我们发现“python-requests/2.19.1”这种数据包含了字母，符号，和数字。于是我们将这些字符，都转换成对应的 ASCII 数值，但同时产生一个问题，如果该字符串过长，会导致最后的换算结果很大，而过短则会很小，影响数据分析。于是这边采用归一化的处理方法。将值限制在（0，1）之间。ip 也可以通过固定的转化规则转化成数值，其他维度方法类似。

3.1.3 归一化

经过数值化的处理后，每条数据格式变成：

1484066360 45 45 1533113323.0 800 0 732 511 302 247 45 2065，这样的数据还是存在问题的，超大的数据会吃掉小数据，小数据在网络传播中会损失其特征，这样训练的模型，很有可能无法达到梯度下降的效果。所以使用常见的归一化过程，针对不同维度数据的特征，采用以下方法：

1) Standardization

Standardization 又称为 Z-score normalization，量化后的特征将服从标准正态分布：

$$z = \frac{x_i - \mu}{\delta}$$

其中，u 和 delta 分别为对应特征的均值和标准差。量化后的特征将分布在[-1, 1]区间。

2) Min-Max Scaling

Min-Max Scaling 又称为 Min-Max normalization， 特征量化的公式为：

$$z = \frac{x_i - min(x_i)}{max(x_i) - min(x_i)}$$

量化后的特征将分布在区间[0, 1]。

针对不同的维度，适当的选取以上两种方法的一种。归一化之后数据格式：

0.61328324457 0.5 0.5 0.8 0.401333333333 0.0298 0.0304 0.7322 0.86 0 1 1538686394.0

3.1.4 打标签

因为我们进行的是监督学习，所以每一条数据都要打上标签。标签格式如下表：

数据类型	标签样式
正常日志 common	1 0
攻击 attack	0 1

表 2 标签格式

在之后，我们会学习更多的攻击方式和防护手段，增加模型可以预测的类型，对不同攻击手段采取不同的处理方法。

3.1.5 加载数据到内存

由于训练数据量比较小，可以一次性直接加载到内存中，加快训练速度。使用以下方法读取数据：

```
def train_reader():
    def reader():
        with open(train,'r') as f:
            lines = [line.strip() for line in f]
            for line in lines:
                line = line.split()
                x = line[0:9]
                y = line[9:11]
                yield x,y
    return reader


def test_reader():
    def reader():
        with open(test,'r') as f:
            lines = [line.strip() for line in f]
            for line in lines:
                line = line.split()
                x = line[0:9]
                y = line[9:11]
                yield x,y
    return reader
```

图 3 数据读取

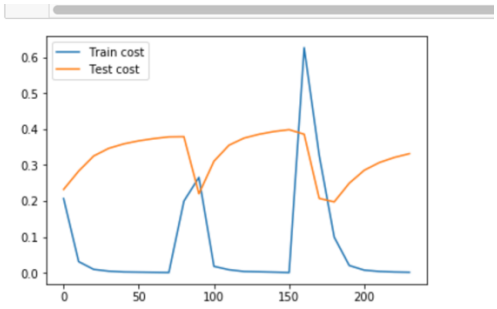
3.1.6 paddlepaddle 训练模型

采用两层神经网络结构训练模型，如下：

```
#定义前向传播
def forward():
    x = fluid.layers.data(name='x',shape=[1,9],dtype='float32')
    hidden = fluid.layers.fc(input=x,size=36,act='relu')
    y_predict = fluid.layers.fc(input=hidden,size=2,act='softmax')
    return y_predict
```

图 4 神经网络结构

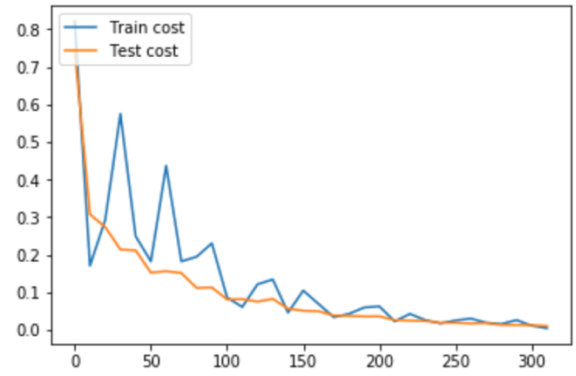
使用 SGD 优化产生一下 cost 图




```
<Figure size 432x288 with 0 Axes>
```

图 5 训练过程（1）

发现 cost 值并不是缓慢下降的，模型无法收敛。调低学习率之后，也不能改变 cost 抖动的问题。采用 adam 优化加入 L2 正则化算法后，继续训练模型，打印出图表，如下：




```
[0.01095960701682738]
```

```
<Figure size 432x288 with 0 Axes>
```

图 6 训练过程（2）

得到了符合预期的损失图。使用该模型进行预测，准确率在 90%左右，查全率在 90%左右，有较好的预测效果的。

3.1.6 paddlepaddle 模型预测

使用以下方法调用 paddlepaddle 模型预测数据：

```
def main():
    inferencer = fluid.Inferencer(
        infer_func=forward,
        param_path=params_dirname,
        place=place)
    tensor_x = np.array(load(data)).reshape(n,9).astype(np.float32)
    results = inferencer.infer({'x': tensor_x})
    #输出结果
    lab = np.argsort(results)
    #格式化输出结果
    i=0
    for ip,time in loading(data):
        ip_out = str(ip) +' '+ str(time)+' '
        print ip,time,
        if(i<=n):
            print lab[0][i]
            ip_out+=str(lab[0][i][-1])
            i+=1
        #将预测结果输出到文件
        with open(data_out,'a') as t:
            t.write(ip_out+'\n')
```

图 7 预测方法

3.2、其他功能实现

该软件除了通过 paddlepaddle 实现恶意访问识别之外，还有人性化的提醒和监控功能。

监控使用 zabbix，可以根据用户定制，选择监控的项目。比如 CPU - 等待 IO CPU 时间比率、CPU - 最近十五分钟服务器负载、网络 - 已建立的 TCP 连接、网络 - TCP 丢包数、内存使用量等。极大的方便了非技术人员对服务器的监控。这里展示，磁盘使用量图、内存使用量图、数据库流量统计图：

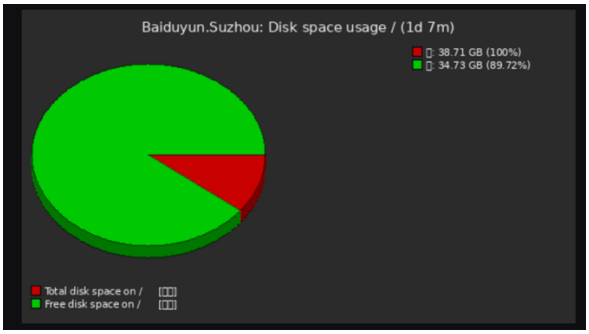


图 8 磁盘容量

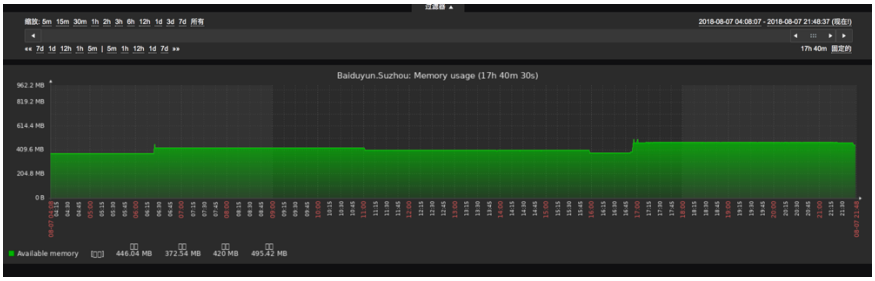


图 9 内存使用

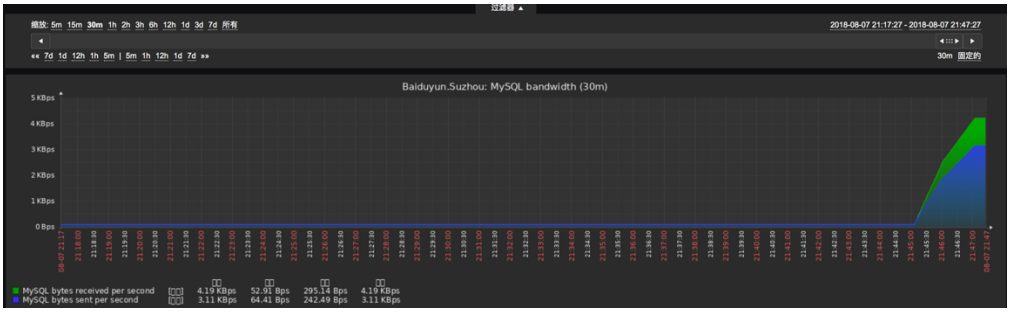


图 10 数据库流量

paddlepaddle 神经网络根据日志信息，预测访问是否为恶意，当识别到恶意访问的用户，系统会自动将恶意识别的 ip 保存到文件中，并且使用 iptables –I INPUT –m set –match-set banthis src –p tcp –destination 80 –j DROP 规则封杀 ip，使得恶意访问用户暂停访问，保护网站安全。

四、应用前景

4.1、行业需求迫切度

近年来，互联网内容和服务市场惊人成长，从政策的利好，到网络环境的优化，再到消费习惯的培养，整体看来，小微型企业以及个人站长的占比也越来越高。而小微企业和个人对中小规模站点的低成本维护的需求，导致对轻量级云防护的需求较大。

4.2、目标客户

小型企业主、小型部门、微商城主、个人站长等都是我们的目标客户。其中个人站长是一个非常特别的群体，通常就是一个人对着一台电脑，做一个网站运营，网站的搭建、技术、内容、推广、广告、安全等全是一个人负责，可以说是样样会，但未必样样都精通，其中最令人头痛的一个问题往往是网站的安全问题，其他的微型企业也会面临这样的烦恼。

本防火墙系统运用 paddlepaddle 对各种常见的攻击行为进行有效识别，并通过集成的机制实时对这些攻击流量进行处理及阻止。

使用本服务器防火墙的优点是，可以减轻非安全专业人员管理服务器的难度，甚至让一个网络新手可以同时管理几台服务器的安全，对于普通的网站漏洞以及轻量级的 CC、DDOS 攻击都能进行拦截和防御。

4.3、市场价值及推广性

本防火墙系统运用 paddlepaddle 对各种常见的攻击行为进行有效识别，并通过集成的机制实时对这些攻击流量进行处理及阻止。使用本服务器防火墙的优点是，可以减轻非安全专业人员管理服务器的难度，甚至让一个网络新手可以同时管理几台服务器的安全，对于普通的网站漏洞以及轻量级的 CC、DDOS 攻击都能进行拦截和防御。

易于上手、稳定可靠的同时减小对系统资源的占用、高性价比、受众广泛。

五、提交材料说明

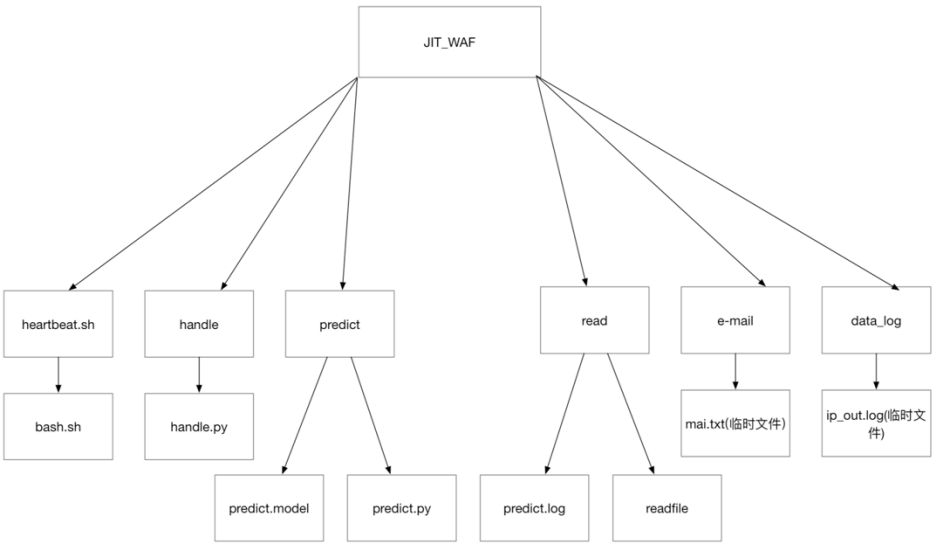
环境说明

PaddlePaddle 版本号为 0.14.0
系统版本为 ubuntu 16.04

5.1、网站搭建

网站搭建过程详细见本人博客
https://blog.csdn.net/jimuka_liu/article/details/81046422

5.2、核心代码目录结构



图一 文件目录树

5.3、文件功能介绍

5.3.1、bash.sh

该文件为核心脚本，是整个软件运行机制的控制中心，保证着项目的稳定运行。启动方法 bash bash.sh，即可启动项目。

5.3.2、handle.py

数据预处理文件，收集到日志文件后，对日志文件进行数据预处理，为神经网络喂入数据做准备。注：不单独启动。

5.3.3、predict.py

PaddlePaddle 预测文件，搭配 predict.model 使用，预测数据，并将产生的结果 ip_out.log 放入 data_log 中。注：不单独启动。

5.3.4、其他文件

这些文件大多为临时文件，用于临时存放数据，配合脚本使用。

5.4、ZABBIX 监控

详细见”2 工程素材/源代码/Zabbix 服务器性能监视平台“说明文档。