《基于Packetbeat和ELK搜集日志》

1. 需求

业务系统会产生各种日志，这些日志有很多用途，其中最主要的用途是：（1）记录用户行为，（2）为定位错误提供依据，保留错误发生时的上下文。一般记录日志的方法就是在代码中埋点；但是，如果只是单纯的记录用户行为，埋点工作就变得非常枯燥。最好是可以不改代码，根据服务器的网络封包自动生成访问日志。

1. Packetbeat特性

Packetbeat是用Go语言编写的，开源。主要分为两个部分：

（1）sniffer：主要用于监听网络端口，抓取数据包；支持三种实现方式pcap、af\_packet、pf\_ring，默认是pcap；支持BPF语法，并可以自动生成BPF表达式，过滤不需要网络封包，也可以在配置文件中配置BPF表达式覆盖自动生成的。

（2）protos：可以根据不同的应用层协议分析、整理、归总网络封包，生成更可读的日志数据；Packetbeat支持常见的协议，比如HTTP、MySql、Redis之类的；甚至可以自己编写协议支持扩展。

Beats（Filebeat、Packetbeat之类的统称）可以直接向Elasticsearch传送日志数据，不需要经过Logstash，除非需要对日志数据进行进一步的过滤和格式化；Beats对Logstash的相性也比Redis、Kafka更好，当Logstash太忙的时候，Beats会放慢发送数据的速度。

Beats可以向Kibana发送日志数据模板（这块还没搞太明白，但是感觉也很牛叉）。

1. 关于ELK的一点补充说明

ELK分别是Elasticsearch、Logstash、Kibana的首字母简称。

Logstash相当于是一个大管道，把从Redis或者Kafka甚至是日志文件中的日志数据发送给Elasticsearch，可以对日志数据进行过滤，格式化，添加附加字段；对于Beats则不再需要，除非之前说的特殊情况。

Elasticsearch有一套开放接口，所以查询数据并不一定要依赖Kibana，推送数据也并不一定要依赖Logstash或Beats。

1. 设计思路
2. 在应用服务器上安装PacketBeat，以抓取网络封包
3. 在PacketBeat配置文件中配置BPF表达式，过滤掉不需要的封包
4. PacketBeat把封包整理成日志数据并发送给Logstash，日志数据中要包含请求和响应的头信息和原数据
5. Logstash进一步过滤掉无意义的日志数据，对过滤后的数据添加应用服务器IP，时间戳等附件字段，并把数据发送给Elasticsearch
6. Elasticsearch+Kibana用于查询数据，也可以利用Elasticsearch提供开放接口开发定制化查询功能。

Logstash

应用（PacketBeat）

应用

Redis

Elasticsearch

Kibana

定制查询后台

1. 问题
2. 就算配置了BPF表达式，PacketBeat仍然会生成一些无意义的日志数据，现在可以将日志数据扔给Logstash再过滤一层，以后可以考虑通过修改源码或者编写新的协议扩展的方式来解决这个问题，这样就不再需要Logstash了。
3. Nginx给浏览器的响应数据是通过gzip压缩的，目前只能在查询后台通过PHP进行gzipdecode，以后可以考虑编写协议扩展来解决。
4. 有些框架会加密Cookie，会对用户身份确认造成一定的麻烦，不过日志数据中一般都会有时间和客户端IP。如果可以的话业务应用最好增加一个不加密的Cookie或者头部信息，用于日志分析，但是就需要在业务应用服务器上做一些小改动了。
5. 具体安装和配置

我懒得写了。。。。。。测试服务器上有 10.70.1.221，文档我也有，感兴趣的同学找我要啊。