

基于 Hbase 数据收集与回溯系统搭建

文档整理：郑志国
邮箱：vista85@sina.com

本文代码位置：

<http://zhuzhibo0.github.io/hbasepacket/>

作者简述：

抓包、存储、回溯无论是在网络排障还是审计等都是可以用到的。由于工作关系，作者所在公司的网络比较复杂，抓包机的部署位置也比较分散，同时流量又很大，比较老套的方式就是一台机器抓包直接存放在本地硬盘，查找的时候用 `tcpdump` 过滤，但是现在都是大数据时代了，所以就尝试用 `hbase` 做了一个所谓的分布式捕获数据包回溯的小工具，用于捕获存储数据包并且可以快速回溯。

软件结构：

1、hadoop-2.6.0、hbase-0.98.12

2、Phb 是抓包程序，phb-service 是提供 web 服务的程序用于查找原始数据包，为了方面编译，相应的软件和插件分别放在了 software 和 plugin 中。

.....


目 录

一、	安装 Ubuntu16.04 LTS	3
1.1	下载地址.....	3
1.2	安装步骤.....	3
1.3	安装软件.....	3
二、	安装 PF_RING 软件包.....	3
2.1	预先安装软件.....	3
2.2	下载代码.....	3
2.3	编译代码与载入模块.....	4
三、	安装与配置 Hadoop	4
3.1	安装 Oracle Java8	4
3.2	创建 Hadoop 的用户访问 HDFS 和 MapReduce	4
3.3	配置 SSH 生成密钥对	4
3.4	关闭 IPv6.....	5
3.5	上传 Hadoop 包.....	5
3.6	更新 Hadoop 配置文件.....	5
3.6.1	更新.bashrc 文件	5
3.6.2	更新 hadoop-env.sh 文件.....	6
3.6.3	更新 core-site.xml 文件.....	6
3.6.4	更新 hdfs-site.xml 文件	6
3.6.5	更新 yarn-site.xml 文件.....	7
3.6.6	更新 mapred-site.xml 文件	7
3.7	格式化节点.....	7
3.8	启动与关闭 Hadoop 进程.....	7
3.9	验证 hadoop 进程	8
四、	安装与配置 Hbase.....	9
4.1	上传 hbse 软件包并解压.....	9
4.2	修改 hbase 文件权限为一般用户权限	9
4.3	设置环境变量.....	9
4.4	配置 hbase-env.sh 文件.....	9
4.5	配置 hbase-site.xml	10
4.6	创建 Hbase 数据库.....	10
4.7	启动 Hadoop 与 Hbase.....	11
五、	安装与配置数据收集与回溯软件.....	11
5.1	安装 Maven 软件包.....	11
5.2	解压软件包.....	11
5.2	将/software 目录中软件解压并安装	11
5.3	编译 phb	12
5.4	编译 phb-service.....	12
六、	测试.....	13
6.1	启用抓包.....	13
6.2	启用回溯工具将流量导出.....	14

一、 安装 Ubuntu16.04 LTS

Ubuntu Server 16.04LTS 是 2016 年最新释放的 Server 版本，作为搭建平台使用的 Linux 操作系统。

1.1 下载地址

进入 <http://releases.ubuntu.com/16.04/> 下载，建议使用 64 位 Bit 版本, 下载地址为 <http://releases.ubuntu.com/16.04/ubuntu-16.04-server-amd64.iso> 

1.2 安装步骤

略,与之前 12.04/14.04 安装过程无区别

1.3 安装软件

系统安装成功后，只需要安装 openssh-server、lrzsz、unzip 软件包

```
$ sudo apt-get install openssh-server
```

```
$ sudo apt-get install unzip
```

```
$ sudo apt-get install lrzsz
```

二、 安装 PF_RING 软件包

2.1 预先安装软件

安装以下软件包，确保软件能够正常编译成功，在 shell 中输入

```
apt-get -y install libpcrc3 libpcrc3-dbg libpcrc3-dev \
build-essential autoconf automake libtool libpcap-dev libnet1-dev \
libyaml-0-2 libyaml-dev zlib1g zlib1g-dev libcap-ng-dev libcap-ng0 \
make flex bison git subversion libmagic-dev pkg-config libnuma-dev
```

如果之前 PF_RING 已经安装则需要卸载，输入 `rmmod pf_ring`

2.2 下载代码

此时直接用 git 工具在网上 download 代码，命令行直接输入

```
git clone https://github.com/ntop/PF\_RING.git
```

2.3 编译代码与载入模块

```
cd PF_RING/kernel/  
make && sudo make install  
cd ../userland/lib  
./configure --prefix=/usr/local/pfring && make && sudo make install  
cd ../libpcap  
./configure --prefix=/usr/local/pfring && make && sudo make install  
cd ../tcpdump  
./configure --prefix=/usr/local/pfring && make && sudo make install  
sudo ldconfig
```

注意，ldconfig 命令需要 root 权限，最后我们会载入该模块，输入：

```
sudo modprobe pf_ring
```

输入成功后检查环境，一切信息是否正常，输入：

```
modinfo pf_ring && cat /proc/net/pf_ring/info
```

三、 安装与配置 Hadoop

3.1 安装 Oracle Java8

```
$ sudo add-apt-repository ppa:webupd8team/java
```

```
$ sudo apt-get update
```

```
$ sudo apt-get install oracle-java8-installer
```

命令执行成功后，程序会安装在/usr/lib/jvm/java-8-oracle 目录，验证程序是否安装正常，执行命令

```
$ java -version
```

3.2 创建 Hadoop 的用户访问 HDFS 和 MapReduce

为了避免安全问题，建议设立新的 Hadoop 用户组和用户帐户来处理所有的 Hadoop 相关的活动。我们将创建的 Hadoop 作为系统组 and 用户由系统用户：

```
$ sudo addgroup hadoop
```

```
$ sudo adduser --ingroup hadoop hduser
```

3.3 配置 SSH 生成密钥对

```
$ sudo su hduser
```

```
$ ssh-keygen -t rsa -P ""
```

```
$ cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
```

3.4 关闭 IPv6

因为 hadoop 只能在 IPv4 协议栈，不支持 IPv6，所以我们禁用 IPv6 功能。

Vi /etc/sysctl.conf 在底部增加以下代码

```
# disable ipv6
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
```

3.5 上传 Hadoop 包

```
$ cd /usr/local/
$ sudo tar -xvzf hadoop-2.6.0.tar.gz
$ sudo mv hadoop-2.6.0 /usr/local/hadoop
$ sudo chown hduser:hadoop -R /usr/local/hadoop
$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode
$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanode
$ sudo chown hduser:hadoop -R /usr/local/hadoop_tmp/
```

3.6 更新 Hadoop 配置文件

3.6.1 更新.bashrc 文件

\$ cd \$HOME //进入当前用户目录，例如当前是 hadoop 用户登录，即比如 cd /home/hadoop

\$ sudo vi .bashrc

```
## Update hduser configuration file by appending the
## following environment variables at the end of this file.
```

```
# -- HADOOP ENVIRONMENT VARIABLES START -- #
export JAVA_HOME=/usr/lib/jvm/java-8-oracle
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
# -- HADOOP ENVIRONMENT VARIABLES END -- #
```

3.6.2 更新 hadoop-env.sh 文件

```
$cd /usr/local/hadoop/etc/hadoop
```

```
$ sudo vi hadoop-env.sh
```

```
## Update JAVA_HOME variable,  
JAVA_HOME=/usr/lib/jvm/java-8-oracle
```

3.6.3 更新 core-site.xml 文件

```
$cd /usr/local/hadoop/etc/Hadoop
```

```
$ sudo vi core-site.xml
```

```
## Paste these lines into <configuration> tag  
<property>  
  <name>fs.default.name</name>  
  <value>hdfs://localhost:9000</value>  
</property>
```

3.6.4 更新 hdfs-site.xml 文件

```
$cd /usr/local/hadoop/etc/Hadoop
```

```
$ sudo vi hdfs-site.xml
```

```
## Paste these lines into <configuration> tag  
<property>  
  <name>dfs.replication</name>  
  <value>1</value>  
</property>  
<property>  
  <name>dfs.namenode.name.dir</name>  
  <value>file:/usr/local/hadoop_tmp/hdfs/namenode</value>  
</property>  
<property>  
  <name>dfs.datanode.data.dir</name>  
  <value>file:/usr/local/hadoop_tmp/hdfs/datanode</value>  
</property>
```

3.6.5 更新 yarn-site.xml 文件

```
$cd /usr/local/hadoop/etc/Hadoop
```

```
$ sudo vi yarn-site.xml
```

Paste these lines into <configuration> tag

```
<property>
```

```
    <name>yarn.nodemanager.aux-services</name>
```

```
    <value>mapreduce_shuffle</value>
```

```
</property>
```

```
<property>
```

```
    <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
```

```
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
```

```
</property>
```

3.6.6 更新 mapred-site.xml 文件

复制 mapred-site.xml 配置模板到目录下

```
$sudo cp /usr/local/hadoop/etc/hadoop/mapred-site.xml.template<空格>
```

```
/usr/local/hadoop/etc/hadoop/mapred-site.xml
```

```
$cd /usr/local/hadoop/etc/Hadoop
```

```
$ sudo vi mapred-site.xml
```

Paste these lines into <configuration> tag

```
<property>
```

```
    <name>mapreduce.framework.name</name>
```

```
    <value>yarn</value>
```

```
</property>
```

3.7 格式化节点

执行命令:

```
hdfs namenode -format
```

3.8 启动与关闭 Hadoop 进程

启动 hdfs 进程:

```
/usr/local/hadoop$ start-dfs.sh
```

启动 MapReduce 进程

```
/usr/local/hadoop$ start-yarn.sh
```

3.9 验证 hadoop 进程

执行 jps 命令

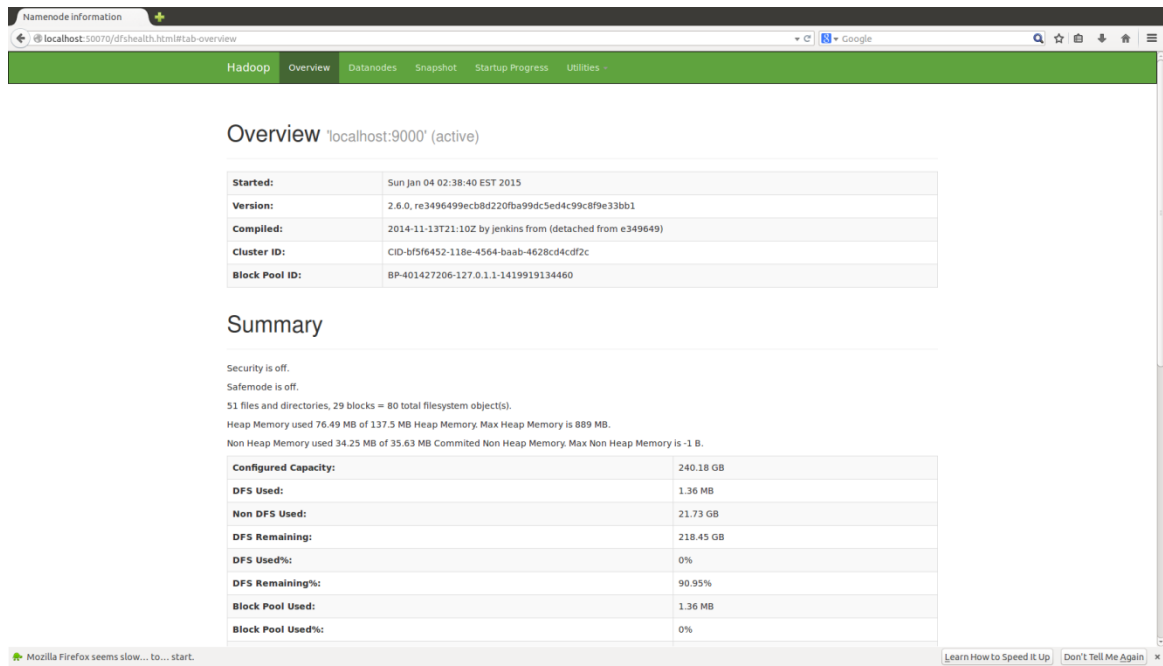
```
$ jps
3472 Jps
3067 SecondaryNameNode
2902 DataNode
3282 ResourceManager
3410 NodeManager
2775 NameNode
```

监控 Hadoop 的 ResourceManage 和 Hadoop 的 NameNode:

ResourceManager – [Http://localhost:8088](http://localhost:8088)

The screenshot displays the Hadoop ResourceManager web interface. The browser address bar shows 'localhost:8088/cluster'. The page title is 'All Applications'. On the left, there is a sidebar with navigation links: 'Cluster', 'About', 'Nodes', 'Applications', 'NEW', 'NEW SAVING', 'SUBMITTED', 'ACCEPTED', 'RUNNING', 'FINISHED', 'FAILED', 'KILLED', 'Scheduler', and 'Tools'. The main content area is titled 'All Applications' and contains a 'Cluster Metrics' table. The table has columns for various metrics: Apps Submitted, Apps Pending, Apps Running, Apps Completed, Containers Running, Memory Used, Memory Total, Memory Reserved, VCoers Used, VCoers Total, VCoers Reserved, Active Nodes, Decommissioned Nodes, Lost Nodes, Unhealthy Nodes, and Rebooted Nodes. The table shows all values as 0. Below the metrics table, there is a section for 'Show 20 entries' with a search bar and a table with columns: ID, User, Name, Application Type, Queue, StartTime, FinishTime, State, FinalStatus, Progress, and Tracking UI. The table is empty, showing 'Showing 0 to 0 of 0 entries'. At the bottom of the page, there is a message: 'Mozilla Firefox seems slow... to... start.' and two links: 'Learn How to Speed It Up' and 'Don't Tell Me Again'.

NameNode – [Http://localhost:50070](http://localhost:50070)



如果能看到显示在上面图片，那么你顺利安装成功了。

四、 安装与配置 Hbase

4.1 上传 habse 软件包并解压

```
$ cd /usr/local  
$ sudo tar xvf hbase-0.98.20-hadoop2-bin.tar.gz
```

4.2 修改 hbase 文件权限为一般用户权限

```
sudo chown -R hduser:hduser hbase-0.98.20-hadoop2
```

4.3 设置环境变量

```
$ sudo vi /etc/profile  
#添加以下内容至文档最后  
export PATH=$PATH:/usr/local/hbase-0.98.20-hadoop2  
保存配置，执行命令  
source /etc/profile
```

4.4 配置 hbase-env.sh 文件

进入 conf 目录，查看是否存相应选项，并将#去掉，编辑正确内容，若不存在该选项，则手

动添加二行，如下：

```
$ cd /usr/local/hbase-0.98.20-hadoop2/conf
$ sudo vi hbase-env.xml
export JAVA_HOME=/usr/lib/jvm/java-8-oracle
export HBASE_MANAGES_ZK=true
```

4.5 配置 hbase-site.xml

```
$ cd /usr/local/hbase-0.98.20-hadoop2/conf
$ sudo vi hbase-site.xml 配置如下内容：
```

```
<configuration>
  <property>
    <name>hbase.rootdir</name>
    <value>hdfs://localhost:9000/hbase</value>
  </property>
  <property>
    <name>hbase.cluster.distributed</name>
    <value>true</value>
  </property>
</configuration>
```

4.6 创建 Hbase 数据库

```
$cd /usr/local/hbase-0.98.20-hadoop2/bin
$sudo ./hbase shell
```

```
2016-07-10 12:28:15,678 INFO [main] Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.available
```

```
HBase Shell; enter 'help<RETURN>' for list of supported commands.
```

```
Type "exit<RETURN>" to leave the HBase Shell
```

```
Version 0.98.20-hadoop2, r9624f3a9eb76f84656a41de0e2099c97f949e831, Tue Jun 7 17:40:20 PDT 2016
```

```
hbase(main):001:0>create 'pcap','t'
hbase(main):001:0>quit
```

注意：如果 create 命令及 list 命令失败，请查看 9000 端口是否正确打开，否则重新启动 hdoop 与 hbase 进程。

4.7 启动 Hadoop 与 Hbase

启用 hadoop: 执行 start-all.sh 命令启动 hadoop 进程, 使用 jps 查看均启动成功;

启用 Hbase:

`$cd /usr/local/hbase-0.98.20-hadoop2/bin`

`$/hbase-daemon.sh start thrift` 再执行 `./start-base.sh` 再次通过 jps 查看进程;



```
$ jps
2992 DataNode
5057 HMaster
5170 HRegionServer
3459 ResourceManager
4995 HQuorumPeer
5256 Jps
2761 NameNode
3262 SecondaryNameNode
3695 NodeManager
```

五、 安装与配置数据收集与回溯软件

进入 <http://zhuzhibo0.github.io/hbasepacket/> 下载最新代码

5.1 安装 Maven 软件包

Maven 是 java 编译软件需要预先安装。

`$sudo bash` #切换到 root 权限

`#apt install maven`

通过执行 `mvn -help` 即可;

5.2 解压软件包

`$cd /home/Hadoop`

`$sudo tar xvf hbasepacket-master.tar.gz`

5.2 将/software 目录中软件解压并安装

`$cd /home/hadoop/hbasepacket-master/software`

`$sudo unzip krakenapps-maven.zip`

`$sudo unzip kraken-master.zip`

`$ sudo unzip glassfish.zip`

`$sudo bash` 切换到 root 权限

```
#cd /home/hadoop/hbasepacket-master/software
#mv glassfish /root/.m2/repository/org/
#mv krakenapps /root/.m2/repository/org/
#mv kraken-master /root/.m2/repository/org/
# cd /root/.m2/repository/org/kraken-master/
#mvs install
```

5.3 编译 phb

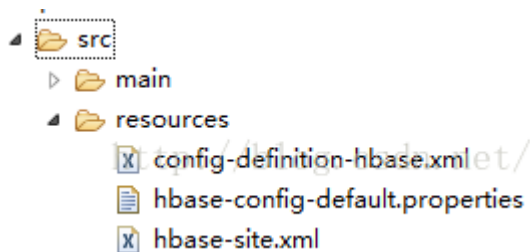
修改 phb/src/main/java/com/hbase/HBaseSender.java 中的
_quorum = "127.0.0.1"; // 可以是列表，格式如：192.168.0.1;192.168.0.2
_port = "2181";然后编译 phb 进行打包 mvn assembly:assembly

```
#cd /home/hadoop/hbasepacket-master/phb
#sudo vi src/main/java/com/hbase/HBaseSender.java
# mvn assembly:assembly
```

5.4 编译 phb-service

1 修改配置文件：

Phb-service 中的 resources 有三个配置文件



第一个 config 不用进行任何修改。

编辑第二个文件：

修改 quorum 为环境相应的 ip 地址以及 port 为相应的 port。

然后修改 hbase.site

修改成和 hbase 环境中同名配置文件一致，ip 地址为 hbase 的 ip 地址。

```
hbase-config-default.properties  hbase-site.xml
1 #hbase zoo keeper configuration
2 hbase.zookeeper.quorum=127.0.0.1
3 hbase.zookeeper.clientPort=2181
4 hbase.client.retries.number=1
5 zookeeper.session.timeout=60000
6 zookeeper.recovery.retry=0
7
8 #hbase table configuration
9 hbase.table.name=pcap
10 hbase.table.column.family=t
11 hbase.table.column.qualifier=pcap
12 hbase.table.column.maxVersions=5
13
14 # scan size limit configuration in MB or KB; if the input is negative o
15 hbase.scan.result.size.unit=MB
16 hbase.scan.default.result.size=6
```

```
hbase-site.xml
1 <?xml version="1.0"?>
2 <?xml-stylesheet type="text/xsl" href="configuration..
5 * Copyright 2010 The Apache Software Foundation
24 <configuration>
25 <property>
26 <name>hbase.zookeeper.quorum</name>
27 <value>127.0.0.1</value>
28 </property>
29 </configuration>
30
```

默认以上为 127.0.0.1 可以不做修改用默认值，以下命令进行编译

```
#cd /home/hadoop/hbasepacket-master/phb-service
```

```
# mvn assembly:assembly
```

六、测试

6.1 启用抓包

```
$cd /home/hadoop/hbasepacket-master/phb/target
```

```
$sudo java -cp phb-1.0-SNAPSHOT-jar-with-dependencies.jar com.zzb.phb 查看你的物理接口，
并在>符后输入你的设备序号回车，开始抓包 30 秒，按 ctrl+c 结束抓包
```

```

root@hadoop:/home/hadoop/hbasepacket-master/phb/target# sudo java -cp phb-1.0-SNAPSHOT-jar-with-dependencies.jar com.
com.zzb.phb.count: 0
com.zzb.phb.readTimeout: 10
com.zzb.phb.snaplen: 65536

log4j:WARN No appenders could be found for logger (org.pcap4j.util.PropertiesLoader).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
NIF[0]: ens33
: link layer address: 00:0c:29:7c:0a:86
: address: /192.168.145.136
NIF[1]: any
: description: Pseudo-device that captures on all interfaces
NIF[2]: lo
: link layer address: 00:00:00:00:00:00
: address: /127.0.0.1
NIF[3]: bluetooth0
: description: Bluetooth adapter number 0
NIF[4]: nflog
: description: Linux netfilter log (NFLOG) interface
NIF[5]: nfqueue
: description: Linux netfilter queue (NFQUEUE) interface
NIF[6]: usbmon1
: description: USB bus number 1
NIF[7]: usbmon2
: description: USB bus number 2
Select a device number to capture packets, or enter 'q' to quit > 0

```

6.2 启用回溯工具将流量导出

```
$cd /home/hadoop/hbasepacket-master/phb/target
```

```
$sudo java -cp phb-service-1.0-SNAPSHOT-jar-with-dependencies.jar
```

```
com.zzb.pcapService.rest.PcapService -port 80
```

此时在打开浏览器，输入

<http://192.168.145.136/pcap/pcapGetter/getPcapsByKeyRange?startKey=0-0-0-0-0-0-0-0&endKey=ffffffff-ffffffff-999-99999-99999-99999-99999> 将数据包导出

附：

1、导包示例

<http://127.0.0.1:8000/pcap/pcapGetter/getPcapsByKeys?keys=0a020a5a-0a20038d-6-22-49795-31905-0&startTime=1454577250386>

<http://127.0.0.1:8000/pcap/pcapGetter/getPcapsByIdentifiers?srcIp=10.2.10.90&dstIp=10.32.3.141&protocol=6&srcPort=22&dstPort=49795>

2、API 说明

索引格式：

0a020a5a-0a20038d-6-22-49795-31905-0

源 ip-目的 ip-协议-端口-srcport-dstport-sessionkey-0

IP 为 16 进制表示格式

GET

pcapGetter/getPcapsByKeys (根据索引)

param:

keys

lastRowKey [一般不用]

startTime [查询开始时间戳 如果为空表示从 0 开始]

endTime [查询结束时间戳 如果为空表示当前时间]

includeDuplicateLastRow [一般不用]

includeReverseTraffic [一般不用]

maxResponseSize [限制返回的 pcap 文件的大小 单位 bytes]

根据索引列表获取数据包:

keys:键值列表 使用','分割

例如:

0a020a5a-0a20038d-6-22-49795-31905-0,0a020a5a-0a20038d-6-22-49795-31906-0

<http://127.0.0.1/pcap/pcapGetter/getPcapsByKeys?keys=0a020a5a-0a20038d-6-22-49795-31905-0,0a020a5a-0a20038d-6-22-49795-31906-0&startTime=1454577250386>

pcapGetter/getPcapsByKeyRange (根据索引范围)

startKey String startKey,

endKeyString endKey,

maxResponseSize String maxResponseSize, [限制返回的 pcap 文件的大小 单位 bytes]

startTime [查询开始时间戳 如果为空表示从 0 开始]

endTime [查询结束时间戳 如果为空表示当前时间]

根据索引范围来获取数据包:

startKey:0a020a5a-0a20038d-0-0-0-0-0

endKey: 0a020a5a-0a20038d-99999-99999-99999-99999-0

以上表示 0a020a5a 到 0a20038d 所有的数据包(10.2.10.90 到 10.32.3.141)

<http://127.0.0.1/pcap/pcapGetter/getPcapsByKeyRange?startKey=0a020a5a-0a20038d-0-0-0-0-0&endKey=0a020a5a-0a20038d-99999-99999-99999-99999-0&startTime=1454577250386>

pcapGetter/getPcapsByIdentifiers(根据特征, 地址, 端口等)

srcIp

dstIp

protocol

srcPort

dstPort

startTime [查询开始时间戳 如果为空表示从 0 开始]

endTime [查询结束时间戳 如果为空表示当前时间]

includeReverseTraffic(默认"false") [一般不用]

根据 ip 地址协议端口等信息获取原始数据包

<http://127.0.0.1:8000/pcap/pcapGetter/getPcapsByIdentifiers?srcIp=10.2.10.90&dstIp=10.32.3.141&protocol=6&srcPort=22&dstPort=49795>