nProbe 安装及基本使用教程

1、简介

nProbe是一款软件NetFlow v5 / v9 / IPFIX探针，使用nProbe能1:1捕获原始流量，并使用标准Cisco NetFlow v5 / v9 / IPFIX格式收集，分析和导出网络流量。nProbe代理模式可以收集sFlow/Jflow/Netstream流数据转化NetFlow v5 / v9 / IPFIX格式输出。它可用于市场上的大多数操作系统（Windows，BSD，Linux，MacOSX）。

2、安装

2.1、在ubuntu 18.04 LTS上安装nProbe

2.1.1、安装存储库

```
sudo apt-get install software-properties-common wget
sudo add-apt-repository universe
sudo wget http://apt-stable.ntop.org/18.04/all/apt-ntop-stable.deb
sudo apt install ./apt-ntop-stable.deb
```

2.1.2、安装软件包

```
sudo apt-get update
sudo apt-get install pfring-dkms nprobe ntopng n2disk cento
sudo apt-get install pfring-drivers-zc-dkms
```

3、在CentOS上安装nProbe

3.1、安装库和依赖项

```
cd /etc/yum.repos.d/
wget http://packages.ntop.org/centos-stable/ntop.repo -O ntop.repo
```

3.2、CentOS/RedHat 8

```
rpm -ivh https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
rpm -ivh http://rpms.remirepo.net/enterprise/remi-release-8.rpm
yum install dnf-plugins-core
dnf config-manager --set-enabled PowerTools
dnf config-manager --set-enabled remi
```

3.3、CentOS/RedHat 7

```
rpm -ivh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

3.4、安装软件包

```
yum erase zeromq3
yum clean all
yum update
yum install pfring-dkms pfring-drivers-zc-dkms n2disk nprobe ntopng cento
```

4、License激活

根据购买的 License，在 License 生成界面填入正确的信息（nprobe -v 获取版本信息和 system ID）。

点击创建License

然后可以看到生成的License

根据系统不同，license 文件的位置不同。这里以 Linux 为例，根据提示复制命令到命令窗口并执行即可自动生成 License。

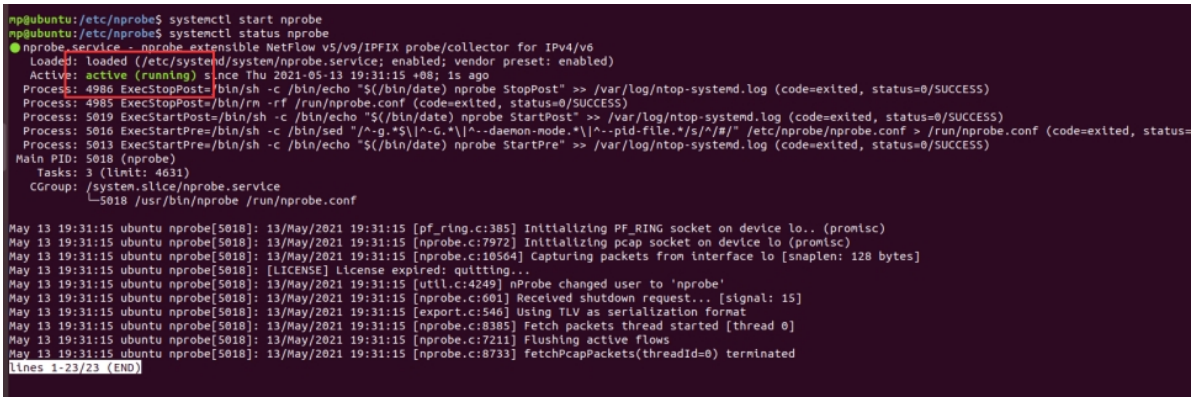
```
root@ubuntu:/home/mp# echo "ArWE9pGculGy6AYBZcCfajvoJSzXHUzjBwZnK3/v7iMSw2mtVhkWLFWFUowDvDPDzbUzg4hk681XYfhlE4HOapCBJf3jl8re488TQeSgtOTIkgDcwi074lVYMPE91bQMp1lSxElKZYGhCyFYm3BBePasyVn060oQUTwghEx/KMzM0Y0ShVKwTikFYmQMnteRReeFcriujgRlS7KUKcxgDuibMDxRiZZ85Gagx1hd+eWbPNU8kv7aWrpSteq+QednxgGh518ty8q1gwfrVlfD9LUuIgkqpPJW0CCdyeV1jQfxRKv8XTmwxQP3a2D+0nuNOxR460KYtu637ZgAZHUC0Q==" > /etc/nprobe.license.gtpv2
```

License 安装完成后，需重新启动 nprobe（`systemctl restart nprobe`），然后使用 `nprobe -v` 查看 License 安装情况。

```
Welcome to nProbe v.9.1.201019 (r6966) for x86_64-pc-linux-gnu
with native PF_RING acceleration.
Copyright 2002-20 ntop.org

Build OS:       Ubuntu 18.04.5 LTS
SystemID:       3B31D45900860F11
GIT rev:        dev:02cc4ab601099b6b2c99ad380b42d6c3b8a76d5e:20201019
Edition:        nProbe Pro
License:        XEgSXvivusC2WR1LbrTzQ2xXDi1kx4Ng5GZbhPgqLT/bIY0I9AiDlUpAdEi9n54rBMv5d
                +jiSag+1cCp3W2o4hJSyo3E9MZUzo5fVnwCzOuyD19UsnBAEZuje9WW3tbLE84WUeDfS8
                mzgBhaNzAP8GiMvUyZg3gs+hkIOB6F2Sx2/OLcKxp1AfyfLmf4mdQtc0c+KSqB3NJIRay
                nLTPE9mExn6+GM5yJWkVx7Xci399MBV+H1cROPbCGq5hwDgxP7VQKcaiRzOrAXpkGrV5q
                Td3wSitni6hXhw02hGC0ZsOr0FbqdTtzcJ11LzLCxC9iRT41MFnp+2ikNH+WV3SoeA==
License Hash:   B3B27F051BB86909F82435900FC3178816040188236D1DA397 [valid license]
License Type:   Time-limited License
Lic. Duration: Until Fri Oct 30 08:47:03 2020 [6 days left]

nProbe is subject to the terms and conditions defined in
the LICENSE and EULA files that are part of this package.

nProbe also contains third party code:
```

5、测试

使用`systemctl start nprobe`启动nprobe，并使用`systemctl status nprobe`查看是否是active状态，最后使用`systemctl stop nprobe`关闭nprobe。



6、nProbe查看帮助

（重要）使用 `nprobe -h` (或 `nprobe -H`)命令查看nprobe使用方法。

7、nProbe典型用例

nProbe可以在三种模式下使用，即：

> - 探针模式
> - 收集器模式（仅流收集，无探针）
> - 代理模式：通过NetFlow接收流并将其（可选地与捕获的流量组合）发送到远程收集器。

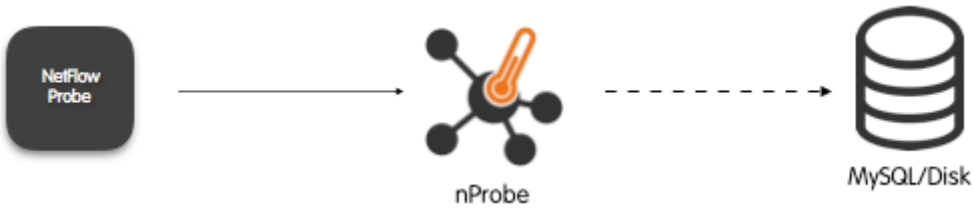
7.1、探针模式


```
nprobe -i eth0 -n collector_ip:2055 -V 10
```

其中：

- -i eth0： 指定捕获流量的接口。
- -n collector_ip:2055 : 指定流收集器地址。
- -V 10： 指定与IPFIX格式输出流。

7.2、收集器模式




```
nprobe -3 2055
```

- -3 2055：从本地2055端口收集netflow/sflow流。

7.3、代理模式


```
nprobe -3 2055 -n collector_ip:2055 -V 9
```

- -3 2055：从本地2055端口收集netflow/sflow流。
- -n collector_ip:2055 : 指定流收集器地址。
- -V 10 ： 指定与IPFIX格式输出流。

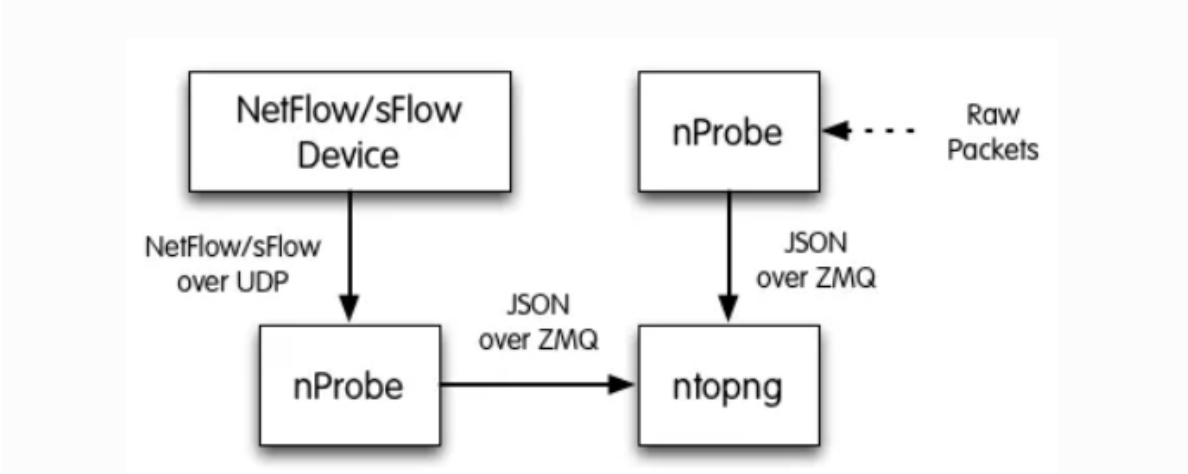
在代理模式下，您可以从NetFlow v5 v9 / NetFlow Lite / sFlow / IPFIX / jFlow转换为NetFlow v5，v9或IPFIX，以便平稳地升级到较新的netflow协议版本，同时利用以前的协议版本。因此，例如，您可以将来自v5路由器的流量转换为IPFIX，反之亦然。请注意，使用某些组合（例如，从v9到v5），您可能会丢失一些流量信息。在这种模式下，收集的流将转换为内部流表示形式，放入高速缓存中，然后根据默认值或-T（如果指定）导出。

8、nProbe和ntopng配合使用示例

ntopng可用于可视化nProbe生成或收集的流量数据。在几种情况下，将ntopng与nProbe一起使用很方便，包括：

- 通常是路由器，交换机和网络设备产生的NetFlow / sFlow数据的可视化。在这种情况下，nProbe从设备收集并解析NetFlow / sFlow流量，并将结果流发送到ntopng以进行可视化。
- 监视连接到远程系统的物理网络接口。在这种情况下，ntopng无法直接监视网络接口，也无法看到其数据包。一个或多个nProbe可用于捕获远程网络接口流量并将结果流发送到中央ntopng，以进行分析和可视化。

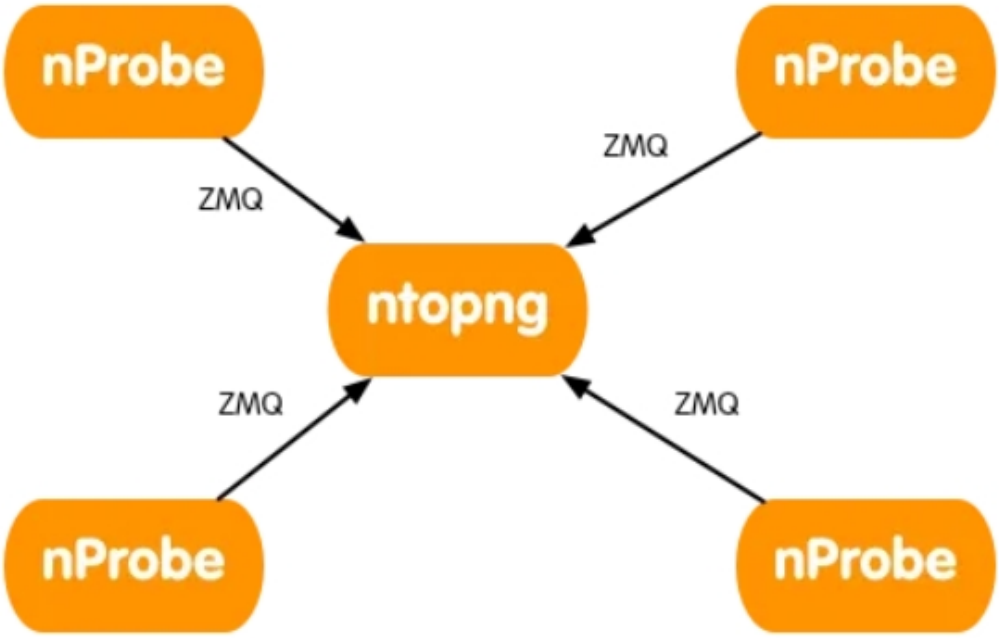
下图总结了上面突出显示的两种情况，并说明了它们也可以组合在一起。



8.1、多个nProbe到ntopng

使用单个ntopng从多个nProbe收集流，可以集中进行流量可视化和存储。

要从多个nProbe收集流，ntopng必须在端口尾部添加c（c表示为收集器），而每个nProbe都需要选择--zmq-probe-mode。在这种配置中，nProbes会启动与充当服务器的ntopng的连接。



以下是这种配置的示例：

```
# ntopng配置
ntopng -i tcp://*:5556c

# nprobe配置
nprobe --zmq "tcp://<ip address of ntopng>:5556" --zmq-probe-mode -i eth1 -n
none -T "@NTOPNG@"

nprobe --zmq "tcp://<ip address of ntopng>:5556" --zmq-probe-mode -i none -n
none --collector-port 2055 -T "@NTOPNG@"

nprobe --zmq "tcp://<ip address of ntopng>:5556" --zmq-probe-mode -i none -n
none --collector-port 6343 -T "@NTOPNG@"
```

9、nProbe流导出示例

nProbe允许您将数据导出到外部源。可以将流导出到ElasticSearch和kafka。以下是所需的配置选项：

```
# ElasticSearch导出命令
--elastic <format>      | Enable export to ElasticSearch

              | Format: <index type>;<index name>;<es URL>;<es user>:<es pwd>

              | Example:

              | --elastic "flows;nprobe-
%Y.%m.%d;http://localhost:9200/_bulk;user:pwd"

              | Note: the <index name> accepts the format supported by
strftime().
```

示例：

```
nprobe -i <interface name> -n none --elastic "flows;nprobe-
%Y.%m.%d;http://localhost:9200/_bulk;user:pwd" -T "@NTOPNG@"
```

获取数据：

```
# 查看ElasticSearch流数据
curl get http://localhost:9200/nprobe***/_search
```

kafka导出选项：

```
--kafka <brokers>;<topic>[;<ack>;<compr>]
              | Send flows to Apache Kafka brokers obtained by metadata
information
              | <host1>[:<port1>],<host2>[:<port2>]... Initial brokers list used
to
              | receive metadata information
              | <topic>    Message topic
              | <0|1|-1>    0=Don't wait for ack
              |        1=Leader ack is enough
              |        -1=All replica must ack
              | <compression> Compression type: none, gzip, snappy
              | Example --kafka localhost;test;0;gzip
```

示例：

```
nprobe -i <interface name> -n none --kafka localhost;test;0;gzip -T "@NTOPNG@"
```

10、编写配置文件

除了使用命令行方式启动外，还可以编写配置文件，以守护进程方式运行nProbe。文件目录为：/etc/nprobe/nprobe.conf，安装完成后会自动生成该文件.文件的编写格式为：option=value(注意等号前后不要随便添加空格)，以-i 命令选项为例，选择捕获流量的接口，可以在配置文件中新的一行添加：-i=eth0 即可eth0为接口名称，一个常用的简单的配置文件示例如下：

```
##############################################
#监控 ens33端口
-i=ens33
-n=none

#输出流到localhost:5556
--zmq="tcp://*:5556"
--zmq-probe-mode=

#输出模板配置
-T="@NTOPNG@"
##############################################
```

启动nprobe守护进程：`systemctl start nprobe`

停止nprobe守护进程：`systemctl stop nprobe`

查看nprobe守护进程：`systemctl status nprobe`

11、常用命令选项

-n： collector addresses

例如：

```
-n 172.22.3.4:33，172.22.3.4:34
-n tcp://172.22.3.4:33
```

这指定nProbe将流发送到的NetFlow收集器地址。如果指定了多个，则需要用逗号将它们分开，或者-n标志可以重复多次（例如，`-n 172.22.3.4:33,172.22.3.4:34`和`-n 172.22.3.4:33 –n 172.223.4:34`）。当定义了多个收集器时，您可以使用–a选项控制流的导出方式（请参见下文）；如果在收集器地址上省略了目标端口，则将流发送到2055端口，而如果未指定所有选项，则默认情况下将流发送到端口2055上的回送接口（127.0.0.1）。使用nProbe导出流向运行在127.0.0.1:2055的收集器。默认情况下，使用UDP协议，但也使用TCP和SCTP（仅在Linux上，当nProbe编译时带有SCTP支持并且内核支持它）。

-i： interface name

例如：

```
-i ens33
```

它指定从中捕获数据包的接口。如果未使用-i，则nProbe将使用默认接口（如果有）。如果用户需要在两个不同的界面上激活nProbe，则他/她需要每个界面一次激活多个nProbe实例。出于调试目的，可以传递nProbe .pcap文件，从中读取数据包。如果nProbe是通过PF_RING支持进行编译和激活的，则可以指定捕获数据包的多个接口。例如，“-i eth0，eth1”会将在eth0和eth1上接收到的数据包合并为一个流量。当合并网络TAP的两个方向（TX和RX）时，此配置特别有用。

-t： maximum flow lifetime

例如：

```
-t 100
```

无论流量持续时间如何，已激活超过指定最大寿命的流量将被视为已过期，并将被释放。属于同一流的其他数据包将在新流中进行说明，单位为秒。

-a： flow export policy

当定义了多个收集器（请参阅–n选项）时，nProbe会将它们的流以循环方式发送。但是，如果使用–a选项，则可以像流重定向器一样向所有收集器发送相同的流。

-b： enable verbose logging

例如：

```
-b
```

使用该标志，nProbe生成可用于调整其性能的详细输出（请参见第2.4章）。零是最低级别（打印很少的信息），1显示流量统计信息，2表示详细信息。流量统计示例：

```
04/Jul/2007 18:16:00 [nprobe.c:1129] Average traffic: [1.7 pkt/sec][1 Kb/sec]
04/Jul/2007 18:16:00 [nprobe.c:1134] Current traffic: [1.9 pkt/sec][1 Kb/sec]
04/Jul/2007 18:16:00 [nprobe.c:1140] Current flow export rate: [0.9 flows/sec]
04/Jul/2007 18:16:00 [nprobe.c:1144] Buckets: [active=13][allocated=21][free=8][toBeExported=0][frag
04/Jul/2007 18:16:00 [nprobe.c:1149] Fragment queue: [len=0]
04/Jul/2007 18:16:00 [nprobe.c:1153] Num Packets: 111 (max bucket search: 0)
04/Jul/2007 18:16:00 [nprobe.c:1170] 115 pkts rcvd/0 pkts dropped
```

-G ： start nprobe as a daemon

例如：

```
-G
```

将nprobe作为守护程序启动。

-P :dump flows

例如：

```
-P ~/Documents
```

该路径指定将流转储到的目录。转储格式是文本，它取决于-T指定的nProbe模板。

-u ： input device index

例如：

```
-u 10
```

NetFlow规范包含一个数字索引，以便标识来自同一探针的不同接口的流。由于可以在同一主机上但在不同设备上启动多个nProbe实例，因此收集器可以使用此标志根据接口号划分流。如果未使用–u，则nprobe将使用0作为接口索引。或者，如果使用-1，则将流发送方的mac地址的最后两个字节用作索引。

-Q : output device index

例如：

```
-Q 11
```

与–u相似，但用于输出接口。

-v : print version

```
nprobe -v
```

该标志用于打印nProbe版本号和日期。

-h : print help

例如：

```
nprobe -h
```

打印nProbe帮助信息。

-H : print help

例如：

```
nprobe -H
```

打印nProbe详细帮助。

-q : host>:[]

例如：

```
-q 127.0.0.1:2033
```

此选项用于指定地址，以及（可选）指定nProbe用来将流发送到以-n表示的目的地的地址。实际上，nProbe将创建一个套接字并将其绑定到:[port]，从而允许用户在离开主机时选择所发出的流所采用的接口。

-T : flow template definition

例如：

```
-T “%IPV4_SRC_ADDR %IPV4_DST_ADDR ...”
```

配置流输出模板，可输出模板类型参考：字段类型

-V : flow export version

例如：

```
-V 9
```

用于指定导出流的流版本。支持的版本是5（v5），9（v9）和10（IPFIX）。

--aggregate-gtp-tunnels

根据隧道ID汇总在每个GTP隧道中流动的流量。

--tunnel

让探针解码隧道流量（例如GTP或GRE流量），从而从此类流量而不是从外部包络中提取流量信息。

-csv-separator : separator

例如：

```
-csv-separator /
```

覆盖默认的“ |” 在转储中使用指定的分隔符。

--dump-metadata : file

例如：

```
--dump-metadata ~/Documents
```

将元数据信息转储到指定文件中并退出。当用户想知道nProbe导出的每个信息元素的类型，以便（例如）他们可以正确地导入数据库时，此选项很有用。

--zmq : socket

例如：

```
--zmq tcp://\*:5556
```

指定一个套接字（例如tcp://*:5556），该套接字将用于向轮询该套接字的订户传递流。可重复使用–zmq命令，如果指定了多个端点，则nProbe使用哈希函数来平衡所有已定义端点之间的流。

--tcp ip:port

例如：

```
--tcp 192.168.0.100:1243
```

通过TCP以JSON格式流向指定的server：port输出。

--db-engine : database engine

如果将流转储到MySQL数据库上（请参阅本手册的后面部分），则nProbe使用的默认数据库引擎是MyISAM。使用此选项，您可以使用其他引擎（例如InnoDB）。

--enable-collection-cache

例如：

```
-i none -3 2055 --enable-collection-cache
```

nProbe实现了流缓存，用于合并属于同一流的数据包。在流收集中，流缓存被禁用。当nProbe在数据包捕获模式下运行时，自动启用流收集缓存。请注意，此选项仅在收集器/代理模式下可用（即使用`-i none`）。

--lua : lua file

例如：

```
--lua lua/dnsSearch.lua
```

调用lua脚本。

--db-engine : database engine

例如：

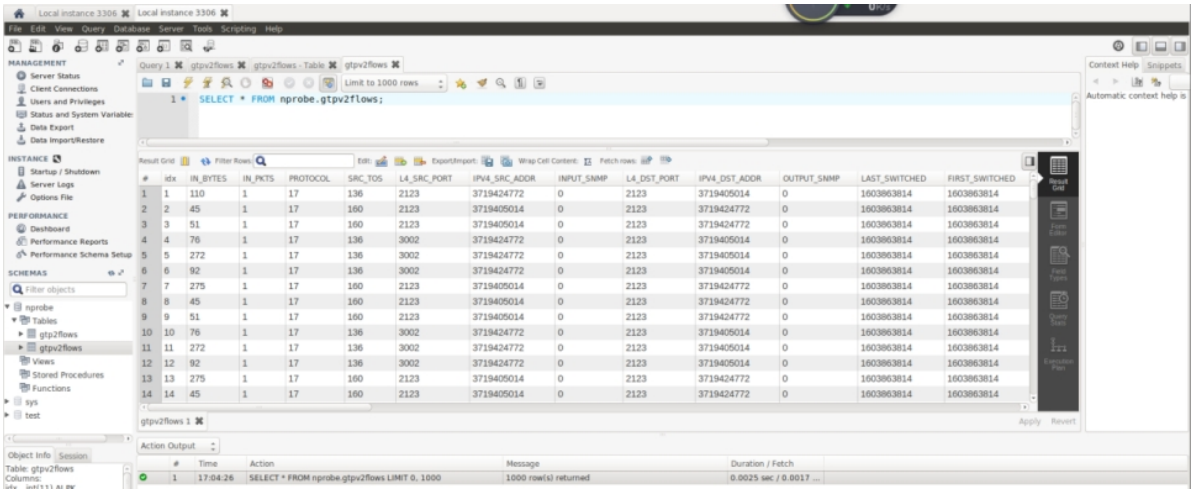
```
--db-engine InnoDB
```

如果将流转储到MySQL数据库上，则nProbe使用的默认数据库引擎是MyISAM。使用此选项，您可以使用其他引擎（例如InnoDB）。

--mysql : host[@port]|unix socket>:dbname:prefix:user:pw

例如：

```
sudo nprobe -i ens33 -n none --db-engine InnoDB --mysql
localhost:nprobe:netflow:root:passwd
```



综合示例：

```
sudo nprobe -i ens33 -n none --db-engine InnoDB --mysql
localhost:nprobe:gtpv2:root:mp1234 -T "%GTPV2_END_USER_IMSI @NTOPNG@"
```

–redis : host[:port]

例如：

```
--redis localhost
```

连接指定的服务器，能够关联GTP-U和GTP-C流量。%FLOW_USER_NAME 使用 IMSI填充。

--cloud

在多个nprobe实例中共享流信息。

12、联系我们

www.hkaco.com 广州| 深圳| 武汉| 成都| 上海| 西安| 北京|台湾|香港 400-999-3848

sales@hkaco.com support@hkaco.com 电话:020-38743030 , 38743032 传真:020-38743233