Table of Contents

7 QoE Stor Module

QoE analytics data collection and storage module

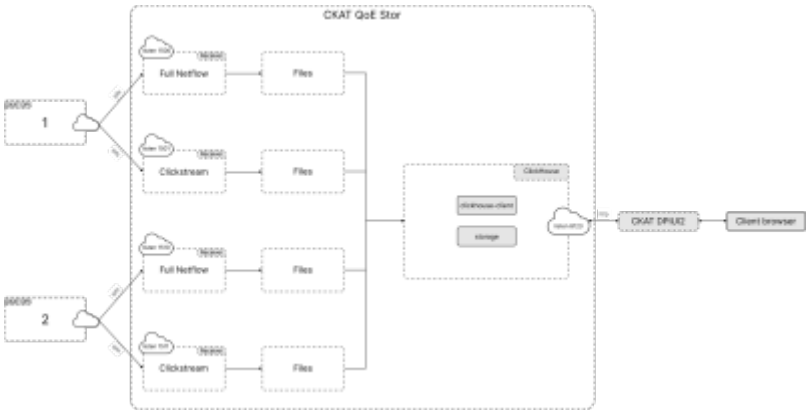
Introduction

The module is designed to collect and store Neflow and Clickstream data. Data is used to analyze QoE in DPIUI2.

Architecture

Data from the VAS Experts DPI is received on several sockets (tcp or udp) using utility designed to collect IPFIX stream data.

The collected data is stored in the ClickHouse database.



Installation and Update

Recommendations on device to be used for installing QoE Stor Module

Do not install the module on the same server with the DPI platform!

Minimum requirements

For the subsystem, you can use hardware or virtual machines with the characteristics listed below:

1. Processor (CPU) 2.5 ГГц - 1 pce
2. RAM - at least 16 GB
3. Hard disk drive (SSD is highly desirable) - at least 500 GB

4. Operating system - CentOS 7+ or CentOS 8+
5. Network interface (NIC) - at least 1 Gbps

Capacity space for QoE and Hardware example

Processor

SSE 4.2 instruction set support is required.

Choose processors with a large number of cores. Clock frequency is less important. For example, 16 cores with 2,600 MHz are better than 8 cores at 3,600 MHz.

Do not disable Hyper-threading and Turbo-Boost.

RAM

RAM should be no less than the amount of data requested.

The more RAM server has, the better performance will be achieved when building reports.

The more memory, the less disk system is stressing.

Minimum prerequisites is 16 GB.

Always disable the swap file.

Disk

Required disk space is at least 16 GB for every storage day, it's actual value depends on daily traffic. It is estimated that 10 Gbit/s of average daily traffic generates approximately 25 GB of data per hour in the QoE Store.

If your budget makes you possible to use SSD, use it. Otherwise use HDD. SATA HDDs 7200 RPM will be suitable.

When using HDD, you can combine them into RAID-10, RAID-5, RAID-6 or RAID-50.

Most of the data is stored in the `/var/lib/clickhouse/` directory. You can mount a drive/partition on this directory.

Temporary data (ipfix dumps) is stored in the `/var/qoestor/backend/dump/` directory. 50 GB be sufficient for this.

Maintenance advice from Yandex ClickHouse

You could familiarize yourself with the contents of maintenance advice from Yandex ClickHouse by following the https://clickhouse.yandex/docs/ru/operations/tips/ link.

Version Information

Version v.1.10.7 (08/05/2021)

- Clickhouse version update to 21.7+

Version v.1.9.0 (12/28/2020)

- Receiving Netstream NAT from third-party systems in QoE Stor. Automatic binding of IP-LOGIN in NAT logs

Version v.1.8.1 (11/12/2020)

- Flexible configuration of fullflow and clickstream aggregation in QoE Stor

Version v.1.7.3 (10/15/2020)

- The ability to store data on HOT and COLD disks
- Reception and storage of NAT logs. The option to generate a NAT log from fullflow. Flexible configuration of the aggregation period and the list of fields to be aggregated
- Reception and storage of GTP logs

Version v.1.6.0 (09/14/2020)

- Option: substitution of the Login field with the value from the vchannel field
- Bug fix

Version v.1.4.2 (06/02/2020)

- Bug fix

Version v.1.4.0 (05/04/2020)

- Clickhouse 20.3 compatibility support
- Ability to install on CentOS 8

Version v.1.3.8 (04/09/2020)

- Updated protocol dictionaries
- Added auto-update of the AS dictionary

Version v.1.3.6 (11/25/2019)

- Updated protocol dictionaries

Version v.1.3.5 (11/06/2019)

- Fixed the behavior of the dictionary of local subnets (got extra addresses)
- Adapted clickstream loading for the correct operation of the user interface in dpiui2-2.8.2

Version v.1.3.4 (10.25.2019)

- Locked in clickstream

Version v.1.3.3 (10/15/2019)

- Update Clickhouse to the latest version (> = 19.15)
- Improved method of loading and processing logs
- Prepared base for the analysis of raw logs
- Connection logs Clear and Netflow

Version v.1.1.1 (09/06/2019)

- Update Clickhouse to the latest version
- Directories asnum_exclude_dic, subnets_exclude_dic and the corresponding filtering mode for these directories

Version v.1.0.9 (21.02.2019)

- Bug concerning incorrect recognition of trunk switches is fixed
- `protocols_dic` dictionary is updated

Version v.1.0.7 hot fixes (24.12.2018)

- ipfix re-export feature is added and is available through the ipfixreceiver configuration file: IPFIX_FULLFLOW_EXPORT and IPFIX_CLICKSTREAM_EXPORT

Version v.1.0.6 hot fixes (04.12.2018)

- Bugs concerning the work of the subnets_local_dic dictionary (such as "A call to function range would produce 12884901882 array elements")
- ipfixreceiver2 receiver configuration has been fixed (FileWriter queue is full. Records dropped.)

Version v.1.0.5 (03.12.2018)

- Host Category Dictionaries
- Transition to the ipfixreceiver2

1. Do not forget to update the installation script (in the Installation section) before upgrading. To upgrade, use the installation script.
2. After the upgrade has been finished, check whether the receivers are running:

   ```
   netstat -nlp | grep 1500
   ```

   and

   ```
   netstat -nlp | grep 1501
   ```

   ipfixreceiver2 should listen corresponding sockets.

3. If receivers are not running, execute the /var/qoestor/backend/qoestor-config.sh script

Version v.1.0.4 (02.11.2018)

- Preaggregation is implemented resulting in reducing the netflow by 6-7 times and clickstream by 3 times
- The following dictionaries are implemented: subscribers, switches, autonomous systems (AS), crc
- The following features are added: identification the traffic direction and subscribers filtering (separation of IP hosts and IP subscribers) by AS and CIDR. This option is meaningful only if the VAS Experts DPI is implemented using the mirror connection scheme.

This QoE Stor version works with the version of DPIUI2-2.1.5 and higher

If you have already installed version 1.0.0, then you should to delete the database due to complete version incompatibility before installing the new version. To do so issue the command:

```
clickhouse-client --query="drop database qoestor"
```

Version v.1.0.0 (20.09.2018)

- A new QoE Stor module is implemented

Installation

Before installing or upgrading, check your Internet connection. Make shure you run scripts under the root or using sudo.

For automatically installation or upgrading follow these steps:

1. Execute fastor-rpm_install.sh script.

   ```
   source <(curl https://vasexperts.ru/install/fastor-rpm_install.sh)
   ```

   It will cause the installation of the following packages: ipfixreceiver, clickhouse, fastor. All of them will be automatically configured according to the defaults.

2. Check whether the qoestor database in clickhouse is available. To do this, issue the command

   ```
   clickhouse-client --query="show databases" | grep qoestor
   ```

3. If there is no database (probably the database server is not running), you should to create it by issuing the following command

   ```
   clickhouse-client -n < /var/qoestor/backend/etc/db/qoestor.sql
   ```

Upgrading

Upgrading is performed using the same scripts as in the installation section.

If receivers stopped after you have executed the

```
yum -y update
```

command, pleas refer to the troubleshooting section following the link.

Configuration

ipfix receivers configuration

ipfix receivers configuration is implemented through the .env file:

```
/var/qoestor/backend/.env
```

Default configuration looks like:

```
#Ipfix form DPI 0
IPFIX_FULLFLOW_PORT_TYPE[0]=tcp
IPFIX_FULLFLOW_PORT[0]=1500
#IPFIX_FULLFLOW_ROTATE_MINUTES[0]=10
#IPFIX_FULLFLOW_ROTATE_DELAY_SECONDS[0]=0
#IPFIX_FULLFLOW_FW_MAX_QUEUE_SIZE[0]=10
#IPFIX_FULLFLOW_DUMP_INSERT_PROCESSES[0]=0
#IPFIX_FULLFLOW_EXPORT[0]=10.0.0.2/9920/tcp,10.0.0.3/3440/udp

IPFIX_CLICKSTREAM_PORT_TYPE[0]=tcp
IPFIX_CLICKSTREAM_PORT[0]=1501
#IPFIX_CLICKSTREAM_ROTATE_MINUTES[0]=12
#IPFIX_CLICKSTREAM_ROTATE_DELAY_SECONDS[0]=400
#IPFIX_CLICKSTREAM_FW_MAX_QUEUE_SIZE[0]=10
#IPFIX_CLICKSTREAM_DUMP_INSERT_PROCESSES[0]=0
#IPFIX_CLICKSTREAM_EXPORT[0]=10.0.0.2/9921/tcp,10.0.0.3/3441/udp

IPFIX_GTPFLOW_PORT_TYPE[0]=tcp
IPFIX_GTPFLOW_PORT[0]=1502
#IPFIX_GTPFLOW_ROTATE_MINUTES[0]=10
#IPFIX_GTPFLOW_ROTATE_DELAY_SECONDS[0]=0
#IPFIX_GTPFLOW_FW_MAX_QUEUE_SIZE[0]=10
#IPFIX_GTPFLOW_DUMP_INSERT_PROCESSES[0]=0
#IPFIX_GTPFLOW_EXPORT[0]=10.0.0.2/9921/tcp,10.0.0.3/3441/udp

IPFIX_NATFLOW_PORT_TYPE[0]=tcp
IPFIX_NATFLOW_PORT[0]=1503
#IPFIX_NATFLOW_ROTATE_MINUTES[0]=10
#IPFIX_NATFLOW_ROTATE_DELAY_SECONDS[0]=0
#IPFIX_NATFLOW_FW_MAX_QUEUE_SIZE[0]=10
#IPFIX_NATFLOW_DUMP_INSERT_PROCESSES[0]=0
#IPFIX_NATFLOW_EXPORT[0]=10.0.0.2/9921/tcp,10.0.0.3/3441/udp

#Traffic direction definition
# 0 - as is
# 1 - by AS (for fullflow only)
# 2 - by CIDR (for fullflow and clickstream)
# 3 - by both: AS and CIDR
# 4 - any: AS or CIDR
TRAFFIC_DIR_DEF_MODE=0

#Subscriber filter
# 0 - no filter
# 1 - by AS (for fullflow only)
# 2 - by CIDR (for fullflow and clickstream)
# 3 - by both: AS and CIDR
# 4 - any: AS or CIDR
SUBSCRIBER_FILTER_MODE=0
```

```
#Subscriber exclude
# 0 - no exclude
# 1 - by AS (for fullflow only)
# 2 - by CIDR (for fullflow and clickstream)
# 3 - by both: AS and CIDR
# 4 - any: AS or CIDR
SUBSCRIBER_EXCLUDE_MODE=0

#Enable host (url) categories dics autoload
URLS_CATEGORIES_DIC_AUTOLOAD_ENABLED=1

#Enable asnum dic autoload
ASNUM_DIC_AUTOLOAD_ENABLED=1

#Enable auto replacing Login with vchannel on insert
# 0 - Disabled
# 1 - Enabled
# 2 - Enabled if Login is empty
ULR_REPLACE_LOGIN_WITH_VCHANNEL=0

# Use dictionary when replacing login
ULR_USE_DIC_WHEN_REPLACING_LOGIN=0

# Enable autoload of vchannel_name_dic
ULR_VCHANNEL_NAME_DIC_AUTOLOAD_ENABLED=0

# vchannel_name_dic remote url
ULR_VCHANNEL_NAME_DIC_URL=

#Import NAT events from fullflow
NAT_IMPORT_FROM_FULLFLOW
# 0 - Disabled
# 1 - Enabled

#Fields to save when aggregating NAT log (bitmask)
# 0x1 - Save protocol ID
# 0x2 - Save event type,
# 0x4 - Save source ipv4,
# 0x8 - Save source port,
# 0x10 - Save destination ipv4,
# 0x20 - Save destination port,
# 0x40 - Save post NAT source ipv4,
# 0x80 - Save post NAT source_port,
# 0x100 - Save session ID,
# 0x200 - Save login,
# 0x400 - Save DPI ID
NAT_AGG_LOG_FIELDS_TO_SAVE_BITMASK=0

#Time interval for aggregating NAT logs
NAT_AGG_LOG_GROUP_TIME_INTERVAL
# 1 - 1 minute
```

```
# 5 - 5 minutes
# 10 - 10 minutes
# 15 - 15 minutes
# 30 - 30 minutes
# 60 - 60 minutes
```

In the configuration above fullflow and clickstream receivers are listening 1500 and 1501 socket respectively. «0» in array subscript means that the receiver get the data from DPI number 0.

It is better to prefer tcp over udp because udp packets can be lost when the MTU is exceeded.

The values TRAFFIC_DIR_DEF_MODE = 0 and SUBSCRIBER_FILTER_MODE = 0 mean that there is no need to calculate the traffic direction and apply filters to subscribers.

The IPFIX_FULLFLOW_EXPORT and IPFIX_CLICKSTREAM_EXPORT parameters allow you to configure export to third-party receivers. Format for use: ip/port/proto[,ip/port/proto.

If the configuration has changed, you should run the /var/qoestor/backend/qoestor-config.sh script

The following example shows how to configure data reception from several DPIs

```
#Ipfix form DPI 0
IPFIX_FULLFLOW_PORT_TYPE[0]=tcp
IPFIX_FULLFLOW_PORT[0]=1500

IPFIX_CLICKSTREAM_PORT_TYPE[0]=tcp
IPFIX_CLICKSTREAM_PORT[0]=1501

#Ipfix form DPI 1
IPFIX_FULLFLOW_PORT_TYPE[1]=tcp
IPFIX_FULLFLOW_PORT[1]=1510

IPFIX_CLICKSTREAM_PORT_TYPE[1]=tcp
IPFIX_CLICKSTREAM_PORT[1]=1511

#Ipfix form DPI 2
IPFIX_FULLFLOW_PORT_TYPE[2]=tcp
IPFIX_FULLFLOW_PORT[2]=1520

IPFIX_CLICKSTREAM_PORT_TYPE[2]=tcp
IPFIX_CLICKSTREAM_PORT[2]=1521
```

The following example corresponds to the situation when you need to identify subscribers by CIDR

This configuration makes sense only when the VAS Experts DPI is installed using port mirroring.

```
TRAFFIC_DIR_DEF_MODE=2
SUBSCRIBER_FILTER_MODE=2
```

Be sure to configure subnets_local_dic dictionary for this configuration example!

The following example corresponds to the situation when export to third-party receivers is configured

```
IPFIX_FULLFLOW_PORT_TYPE[0]=tcp
IPFIX_FULLFLOW_PORT[0]=1500
IPFIX_FULLFLOW_EXPORT[0]=10.0.0.2/1600/tcp

IPFIX_CLICKSTREAM_PORT_TYPE[0]=tcp
IPFIX_CLICKSTREAM_PORT[0]=1501
IPFIX_CLICKSTREAM_EXPORT[0]=10.0.0.2/1601/tcp
```

Restarting the receivers

All receivers can be restarted using the command:

```
/var/qoestor/backend/qoestor-config.sh
```

If you need to restart the receivers one by one, you can do this by restarting corresponding systemd service units, for example

- For CentOS 7

  ```
  systemctl restart qoestor_fullflow_0.service
  systemctl restart qoestor_clickstream_0.service
  ```

- For CentOS 6

  ```
  service qoestor_fullflow_0 stop
  service qoestor_clickstream_0 stop
  service qoestor_fullflow_0 start
  service qoestor_clickstream_0 start
  ```

Stoping the receivers

- For CentOS 7

  ```
  systemctl stop qoestor_fullflow_0.service
  systemctl stop qoestor_clickstream_0.service
  ```

- For CentOS 6

  ```
  service qoestor_clickstream_0 stop
  ```

```
service qoestor_fullflow_0 stop
```

Clickhouse DB stop and start

- Stop

  ```
  sudo /etc/init.d/clickhouse-server stop
  ```

- Start

  ```
  sudo /etc/init.d/clickhouse-server restart
  ```

DPI configuration

Export configuration

The DPI version must be at least 8.1.

You can configure ipfix export by editing the fastdpi.conf configuration file on your DPI device.

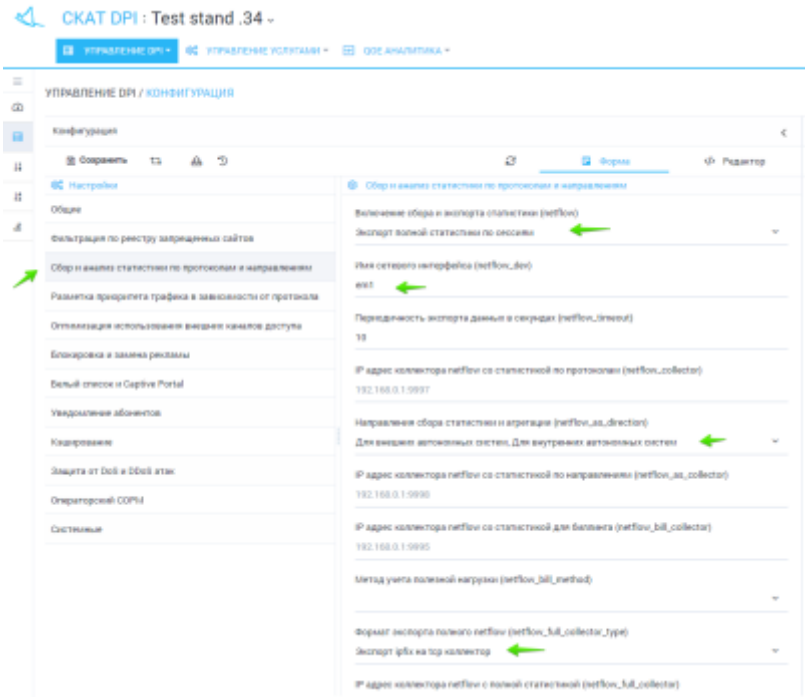
```
netflow=8
netflow_dev=em1
netflow_timeout=10
netflow_as_direction=3
netflow_full_collector_type=2
netflow_full_port_swap=0
netflow_full_collector=YOUR_QOESTOR_IP:1500
ipfix_dev=em1
ipfix_tcp_collectors=YOUR_QOESTOR_IP:1501
```

fastdpi restart is needed for the changes to take effect:
service fastdpi restart

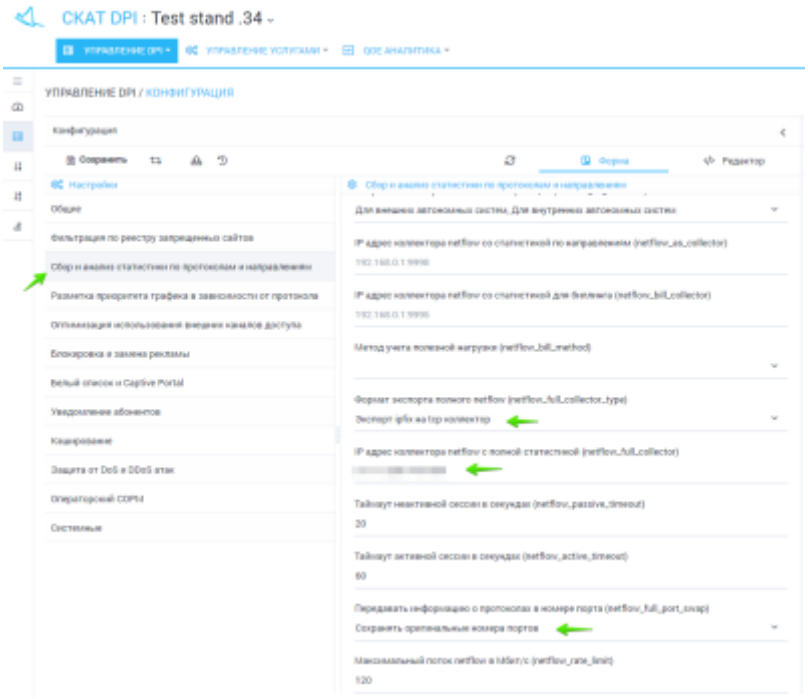
This can be achieved also using DPIUI2.The dpiui2 version must be at least 2.1.0.

To perform configuration using DPIUI2 you should open the section DPI CONTROL → CONFIGURATION. Open the tab **Collection and analysis of statistics on protocols and directions**.

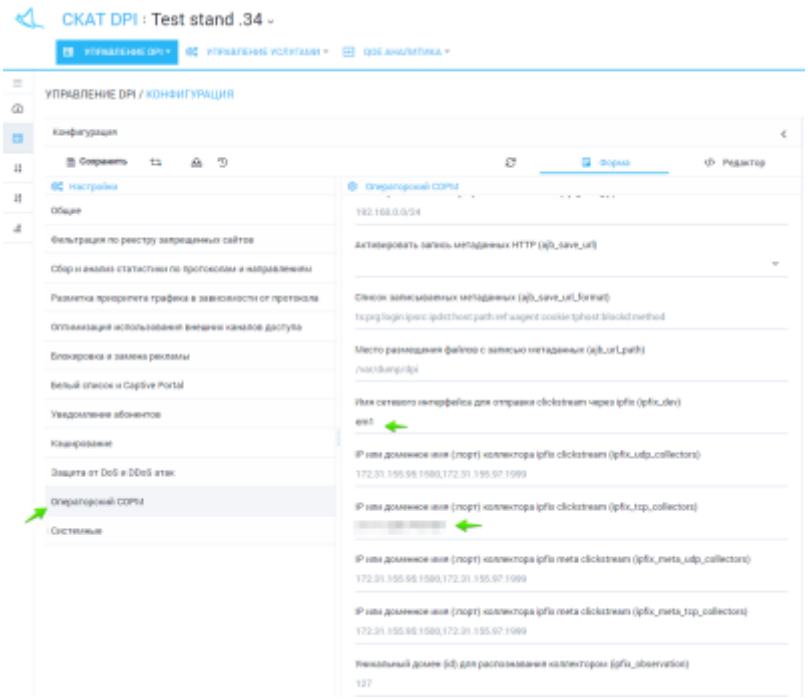
Set neflow option to **Export of complete statistics for sessions**. See figure below.



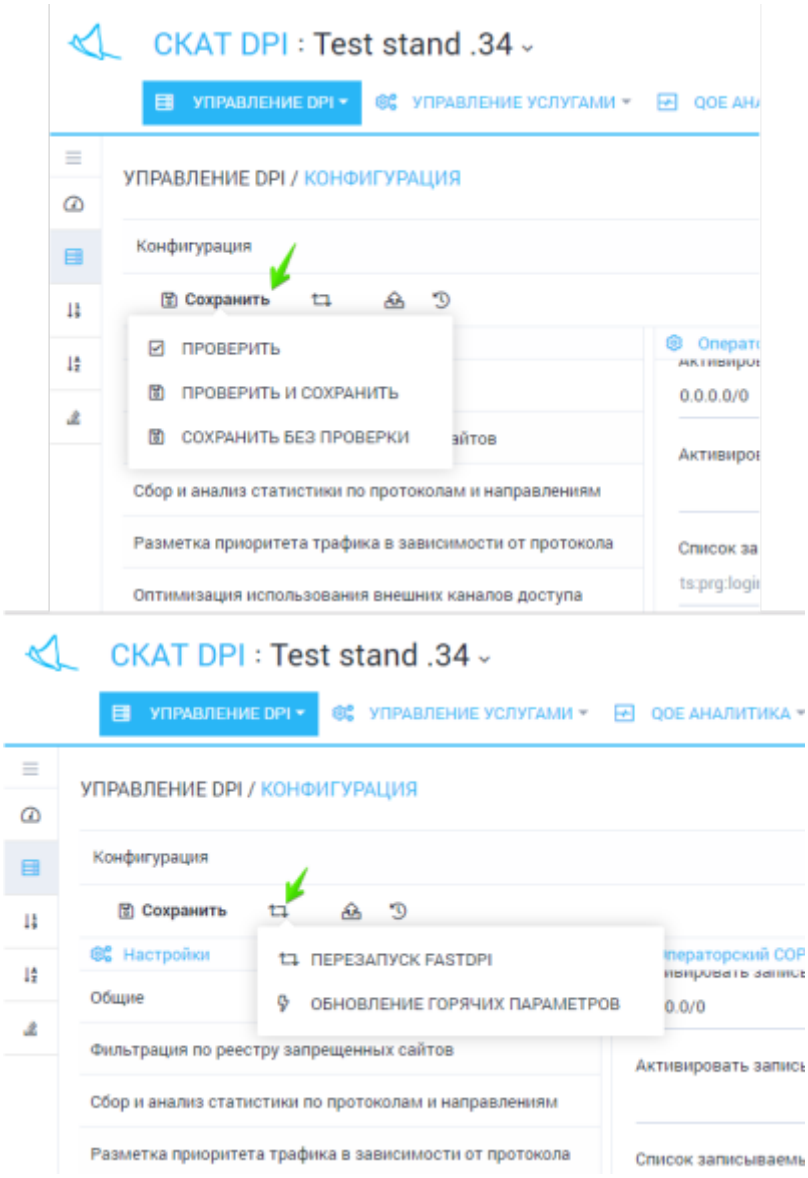
Then specify socket for fullflow receiver within "netflow_full_collector" parameter: "IP address of the netflow collector with full statistics (netflow_full_collector)". "netflow_full_collector_type" should be set to "Export ipfix to udp header", whereas "netflow_full_port_swap" should be empty or equals to "Keep original port numbers". See the figure below.



Type in the clickstream receiver socket in "ipfix_udp_collectors" parameter under the "DPI CONTROL → CONFIGURATION → Operator's SORM → IP or the domain name (: port) of the ipfix collector clickstream (ipfix_udp_collectors)". See the figure below.

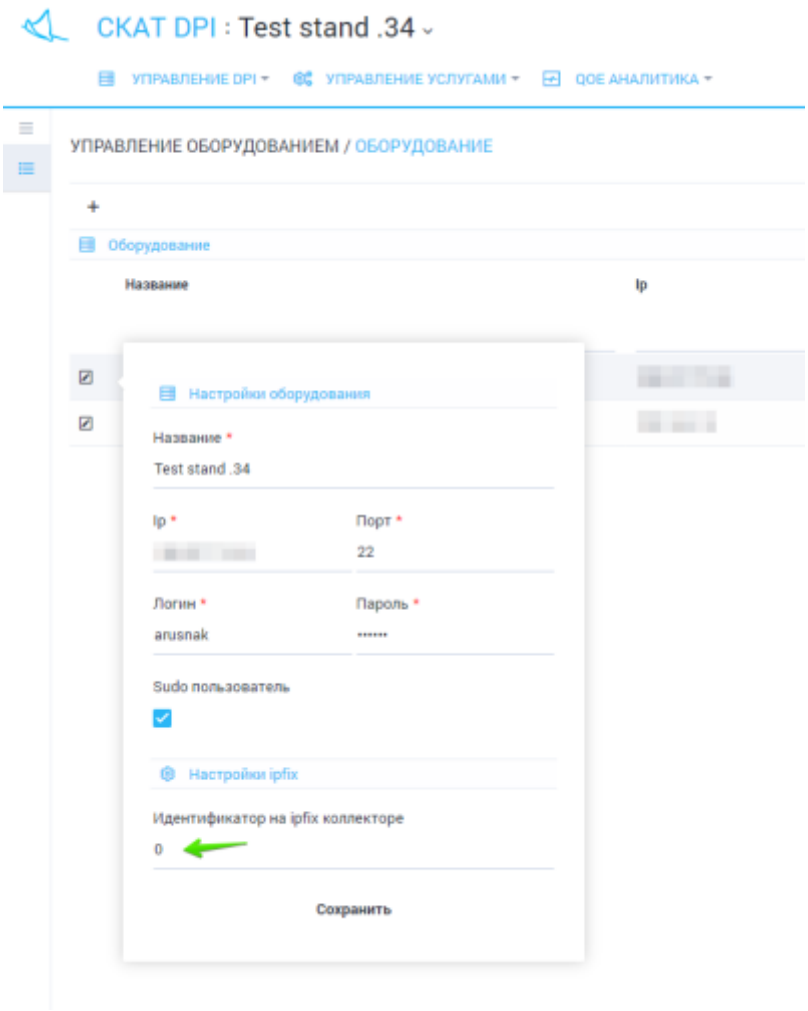


Press Save and then restart fast_dpi. See the figure below.



ID assignment to ipfix collector

Open the section Hardware "Management → Equipment → Hardware settings". Type in an identifier within the "Id on ipfix collector" section (under the "Ipfix settings") for the ipfix collector. See the figure below.



Setting up DPIUI2 connection to the QoE Stor module

In order to browse QoE reports, you should configure the DPIUI2 connection to the QoE Stor. See the Setting up a connection to the QoE Stor section.

Configuring the dictionaries

All dictionaries are in the /var/qoestor/backend/etc/db/ directory and have .txt extension

Each dictionary has it's own sample.txt which can be used as a pattern.

All columns within the dictionaries are separated by a tab character (\t). The number of \t should be one less than the number of columns in dictionary. Please, pay close attention to this circumstance.

When files are changed, the data is loaded into the database automatically.

Some useful commands when working with dictionaries:

- Speed up data updates in directories

```
clickhouse-client --database=qoestor --query="system reload
dictionaries"
```

- Check dictionaries for errors

```
clickhouse-client --database=qoestor --query="select * from
system.dictionaries"
```

- Check if there is data in the directory, for example for the `subnets_local_dic`

```
clickhouse-client --database=qoestor --query="select * from
subnets_local_dic"
```

asnum_local_dic and subnets_local_dic dictionaries

The lists of your local AS and local subnets are specified in these dictionaries. Dictionaries are used to identify the traffic direction (true in case the DPI is installed using mirroring) and to filter subscribers (so that the hosts IP addresses do not appear in the subscriber reports)

Example of **asnum_local_dic** dictionary

```
12345   LOCAL
65535   UNKNOWN
```

The first column is AS number, the second one is it's name (it is displayed in reports).

Example of **subnets_local_dic** dictionary

```
192.168.1.0/24  LOCAL
10.64.66.0/24   LOCAL
172.16.0.0  LOCAL
2a02:2168:aaa:bbbb::2   LOCAL
```

The first column is IP address or CIDR, the second one is the name (it is NOT displayed in reports, but it is required by format).

Do not add too large subnet. Break into small ones. Highest value (limit) is 100000000

subscribers_dic, switches_dic, crc_dic dictionaries

subscribers_dic

Dictionary of subscribers.

Dictionary example

```
10.64.66.100    login    5    port1    unit_vendor    cabel    contract
services    mac
10.64.66.101    login    2    port1    unit_vendor    cabel    contract
```

```
services    mac
10.64.66.102    login    3    port1    unit_vendor    cabel    contract
services    mac
10.64.66.103    login    4    port1    unit_vendor    cabel    contract
services    mac
10.64.66.104    login    5    port1    unit_vendor    cabel    contract
services    mac
10.64.66.105    login    5    port2    unit_vendor    cabel    contract
services    mac
10.64.66.106    login    5    port3    unit_vendor    cabel    contract
services    mac
```

Columns:

1. IP address
2. Login
3. Switch ID (access switch)
4. Switch port
5. Subscriber device vendor
6. Cable
7. Contract
8. Services
9. Subscriber device MAC address (is reserved for future purposes)

switches_dic

Hierarchical dictionary of devices (access switches and trunk switches)

Dictionary example

```
1   Switch_1    Ethernet    Region_1    Address_1    10.140.1.18    ISP_1
0    0
2   Switch_2    Ethernet    Region_2    Address_2    10.140.2.18    ISP_1
0    0
3   Switch_3    Ethernet    Region_3    Address_3    10.140.3.18    ISP_1
0    1    port1
4   Switch_4    Ethernet    Region_4    Address_4    10.140.4.18    ISP_1
0    3    port1
5   Switch_5    Ethernet    Region_5    Address_5    10.140.5.18    ISP_1
0    4    port1
```

Columns:

1. Device ID UInt64
2. Device name
3. Device type
4. Region
5. Address
6. Switch IP address
7. Internet service provider

8. Indicator: trunk switch indicator (1 if so). Is not currently used, you can set 0 everywhere
9. Upstream Switch ID UInt64
10. Upstream Switch port
11. The owner

crc_dic

CRC Errors Dictionary (on switch ports)

Dictionary example

```
2   port_1    450
5   port_1    550
5   port_2    500
4   port_1    780
```

Columns

1. Switch ID
2. Switch port
3. CRC value

urlcats_dic and urlcats_host_dic dictionaries

Host categories dictionaries. Designed to determine the ownership of a particular host category.

Directories are automatically downloaded from vasexperts.ru resources.

To speed up the initial load, issue the following commands

1. ```
   sh /var/qoestor/backend/etc/cron_daily.sh
   ```

2. ```
   clickhouse-client --database=qoestor --query="system reload
   dictionaries"
   ```

Transferring dumps and database to a separate drive

All data is stored in the section /var by default.

For example you have a separate drive connected to /home.

1. Work from the root user

   ```
   sudo su
   ```

2. Stop the recievers and DB

```
systemctl stop qoestor_fullflow_0.service
systemctl stop qoestor_clickstream_0.service
sudo /etc/init.d/clickhouse-server stop
```

3. Create dictionaries in the /home section

```
mkdir /home/qoestor
mkdir /home/qoestor/clickhouse
mkdir /home/qoestor/dump
```

4. Copy tha data on the new drive

```
cp -r /var/lib/clickhouse/* /home/qoestor/clickhouse
cp -r /var/qoestor/backend/dump/* /home/qoestor/dump
```

5. Change the folder owner /home/qoestor/clickhouse

```
chown -R clickhouse:clickhouse /home/qoestor/clickhouse
```

6. Delete the old dictionaries

```
rm -rf /var/lib/clickhouse
rm -rf /var/qoestor/backend/dump/
```

7. Create simlinks

```
ln -s /home/qoestor/clickhouse /var/lib/clickhouse
ln -s /home/qoestor/dump /var/qoestor/backend/dump
```

8. Check the links

```
readlink -f /var/lib/clickhouse
readlink -f /var/qoestor/backend/dump
```

9. Run the DB

```
sudo /etc/init.d/clickhouse-server restart
```

10. Start the recievers

```
sudo sh /var/qoestor/backend/qoestor-config.sh
```

Troubleshooting

QoE Stor module does not work, although everything was installed according to the instructions.

If you have installed and configured everything according to the instructions above, and the DPIUI2 "QoE Analytics" section is empty, below is a checklist of the steps to be taken before contacting our technical support:

1. Check the time and timezone settings on servers with dpiui2 and QoE Stor installed. Try to specify a long period in dpiui2. If it's about the timezone, the data will appear. Set the proper time on the servers with dpiui2 and QoE Stor module installed , restart the corresponding servers (on which the dpiui2 and QoE Stor module are installed).
2. Check if the database is created on the server with QoE Stor installed

   ```
   clickhouse-client --query="show databases" | grep qoestor
   ```

   If the database is not created, you should create it using the following command

   ```
   clickhouse-client -n < /var/qoestor/backend/etc/db/qoestor.sql
   ```

3. Check if there is data in the database on the server with QoE Stor installed

   ```
   clickhouse-client --query="select count(), min(flow_start_time),
   max(flow_start_time) from qoestor.fullflow"
   ```

   и

   ```
   clickhouse-client --query="select count(), min(time), max(time) from
   qoestor.clickstream"
   ```

4. Check the content of the receiver dumps on the server with QoE Stor installed

   ```
   /var/qoestor/backend/dump/fullflow
   ```

   and

   ```
   /var/qoestor/backend/dump/clickstream
   ```

5. Check the receivers logs under the

   ```
   /var/qoestor/backend/logs/
   ```

   directory. Is there something like "oops!”? Please contact technical support, because there is most likely the components are not installed from the proper repositories". If there is "Illegal IPFIX Message Version 0x0005", then once again check the export settings on the dpi: `netflow_full_collector_type` is specified improperly.

6. Check whether the 1500 and 1501 ports are listening on the server with QoE Stor installed

   `netstat -nlp | grep 1500` и `netstat -nlp | grep 1501`

   Restart all the receivers, just to be safe, by issuing the command:

   ```
   /var/qoestor/backend/qoestor-config.sh
   ```

7. Check again ipfix export settings on the dpi device
8. Check the GUI connection details for QoE Stor on the server with DPIUI2 installed
9. Check if the ClickHouse database is running on the server with QoE Stor installed by issuing following command:

```
ps aux | grep clickhouse
```

Make sure the server has enough amount of RAM.

10. Check the clickhouse logs under the /var/log/clickhouse-server/ directory on the server with QoE Stor installed.

If there is a need to drop all data in the database, then you should take the following steps on the server with the QoE Stor installed:

1. Drop the DB by issuing:

```
clickhouse-client --query="drop database qoestor"
```

2. Recreate DB by issuing

```
clickhouse-client -n < /var/qoestor/backend/etc/db/qoestor.sql
```

yum –y update command is issued, but receivers are still not running

When running **yum –y update** some libraries can be broken resulting in receivers are not running. Follow these steps to troubleshoot:

1. Remove fastor and its dependencies

```
 yum remove fastor ipfixreceiver libfixbuf netsa_silk netsa-python
```

2. Reinstall it using fastor-rpm_install.sh.gz script.

SQL and data uploading using CSV, JSON, TabSeparated formats

If necessary, you can create your own reports without additional tools and upload data in any format such as CSV, JSON, TabSeparated.

Data is stored in 4 main logs:

- qoestor.fullflow – full netflow log, storage period is 24 hours
- qoestor.clicksteam – full clickstream log, storage period is 24 hours
- qoestor.fullflow_agg – pre-aggregated neflow log, unlimited storage period
- qoestor.clicksteam_agg – aggregated clickstream log, unlimited storage period

Use the following command format

```
clickhouse-client --database=qoestor  --query="Your_SQL_query"
```

Data is uploaded using TabSeparated format by default.

Example. The client asked for a log of connections to a specific host in CSV format.

```
clickhouse-client --database=qoestor --query="select * from fullflow
prewhere flow_start_date = '2018-10-04' where (source_ipv4 = '10.64.66.100'
or destination_ipv4 = '10.64.66.100') and host = 'google.com' ORDER BY
flow_start_time limit 10 format CSV"
```

For more information on ClickHouse SQL, follow the link:
https://clickhouse.yandex/docs/ru/query_language/select/.