Содержание

Questions and answers

Why an address pool of at least 2 or 4 addresses is recommended to create?

Lock free scheduling algorithm implemented in the DPI is designed to distribute sessions over the working threads, so it imposes restrictions on the public IP address which can be assigned to subscriber from the pool:

- It is required that the number of addresses within the pool is at least the number of working threads (2 for the Stingray SG-6 and 4 for the Stingray SG-10 and further) in order to ensure that public address is assigned to subscriber.

To find out the number of working threads flows:

```
expr $(ps -p `pidof fastdpi` H -o comm|grep wrk|wc -l) / $(ps -p `pidof fastdpi` H -o comm|grep rx|wc -l)
```

- If there is the only address in the pool, then the address can be assigned just for those subscribers that are used by balancing algorithm, not all subscribers.

How to determine which public address from the pool the subscriber will receive?

To see which public address was assigned to a private one, you can use the command

```
fdpi_ctrl list status --service 11 --ip 192.168.4.20
```

In NAT 1: 1, the public address is allocated immediately when the service is assigned, in CG-NAT at the time of the session start

Also, the public address allocated to the subscriber is reported to Radius Accounting for the purpose of logging it in billing.

It is impossible to predict in advance which address will be issued to a subscriber from the pool: it depends on various factors and, in particular, on the current load of the pool.

Idle (inactive) SSH sessions began to get disconnected after enabling NAT

Indeed, the NAT session lifetime is limited, since the subscriber sessions number is a limited resource and a large number of idle (inactive) sessions in the pool reduces NAT performance and consequently the total performance.

NAT doesn't have the ability to distinguish whether the session was terminated abnormally or is simply inactive, so NAT closes such sessions because inactivity timeout has been exceeded. Such a behavior is provided by the standard and is supported by most CG-NAT vendors.

Sessions lifetime in Stingray SG can be configured by followin settings

```
lifetime_flow=60
lifetime_flow_long=600
```

where lifetime_flow_long is a lifetime in seconds of inactive TCP-sessions, lifetime_flow regards the remaining TCP-sessions.

The values of reviewed above settings should not be too high, since it can cause CG-NAT performance reduction due to enormous session table, also it can cause the subscriber session limit being exhausted (is set by nat pool settings).

Therefore, it is recommended to use tcp keep-alive mechanism when the long-running inactive connections take place, it means that the empty packet will be sent regularly within the session which indicates the session still active.

You can configure tcp keep-alive either application-wide on the server or client side, or operating system-wide at once.

SSH server setting example

```
in /etc/ssh/ssh_config file add the following line
ServerAliveInterval 60
```

SSH clien-side setting example

```
in ~/.ssh/config file add the following lines
Host *
  ServerAliveInterval 60
```

```
or using terminal
ssh -o TCPKeepAlive=yes -o ServerAliveInterval=60 user@example.com
```

System-wide setting example for the CentOS

```
in /etc/sysctl.conf file add the following lines
net.ipv4.tcp_keepalive_time = 600
net.ipv4.tcp_keepalive_intvl = 60
net.ipv4.tcp_keepalive_probes = 20
```

How many private IP addresses can be hidden behind the

public one in CGNAT?

It is recommended to maintain a ratio of 1:10 (better) to 1:100 (worse), although you can hide even a thousand of private IP addresses behind the public one.

Details:

64000 ports are available on one public IP for CGNAT by default, each port corresponds to one tcp and one udp session. The number of sessions created by different groups of subscribers differs: individuals create fewer sessions, legal persons create more (therefore, you need to use a separate pool with different limits on the sessions number in case of legal persons); note that a subscriber with a torrent can create up to 1000 sessions in peak.

On average, an individual creates 50-60 concurrent sessions, i.e. 64000/60 = 1066 individuals can be hidden behind one private IP, but in practice such a significant oversubscription is not recommended, since many popular services (such as mail, video, search) use protection against botnet attacks networks based on IP addresses, so if they receive too many requests from one address, they consider it as an attack and block some requests or enable captcha, which in turn will create inconvenience to subscribers.

It is also necessary to take into account the specialty of the port release mechanism in the NAT Pool:

1. When Service 11 is connected, the subscriber is assigned Public IP based on the distribution algorithm
2. When the subscriber starts to establish sessions, the ports are taken from the general Stingray SG queue and assigned with certain timeouts
3. If there are many subscribers on a specific Public IP who start competing for free ports, subscribers may start experiencing access problems.

Recommendations for NAT Pool creation and operation:

1. The blocked subscribers (Service 5 + policing) should be placed in a separate NAT Pool so that they do not affect the work of active subscribers. This is how the IPhone behaves, for example: it establishes many sessions while searching for a working service
2. Create sparse pools and separate clients into different NAT Pool by type: Individuals and Legal entities
3. Monitor and work with clients who create a heavy load. To receive, process and store NetFlow with DPI, we suggest using a software product for collecting statistics QoE Stor and a DPIUI2 graphical interface. You will be able to analyze the subscriber's traffic and conclude that his PC is infected.

How to change the parameters of an existing and being used pool?

1) To change the limit on a number of sessions:

```
fdpi_ctrl load profile --service 11  --profile.name test_nat_2000 --
```

```
profile.json '{ "nat_ip_pool" : "111.111.111.0/24", "nat_tcp_max_sessions" :
2000, "nat_udp_max_sessions" : 2000, "nat_type" : 0 }'
```

The command to create a pool is identical to the previous one but with different `nat_tcp_max_sessions` and `nat_udp_max_sessions` settings

2) To add the additional addresses to the pool:

```
fdpi_ctrl load profile --service 11  --profile.name test_nat_2000 --
profile.json '{ "nat_ip_pool" : "111.111.111.0/24,222.222.222.0/25",
"nat_tcp_max_sessions" : 2000, "nat_udp_max_sessions" : 2000, "nat_type" : 0
}'
```

The command to create a pool is identical to the previous one with an additional pool specified with a comma.

3) To reduce the pool

The current version does not support pool size dynamic reduction and deletion addresses from it. To do so you should free the pool, delete it and create it with new parameters.

Please install jq (a utility for working with data in JSON format) for your convenience:

```
yum install epel-release yum-utils
yum-config-manager --disable epel
yum --enablerepo epel install jq
```

After that we will save information about the subscribers of the current pool, delete pool, create one and assign the subscribers to it:

```
fdpi_ctrl list all --service 11 --profile.name test_nat_4000 --outformat
json|jq '.lservices[] | .login | select(. != null)' > save_users.txt
fdpi_ctrl list all --service 11 --profile.name test_nat_4000 --outformat
json|jq -r '.lservices[] | .ipv4 | select(. != null)' >> save_users.txt
fdpi_ctrl del all --service 11 --profile.name test_nat_4000
fdpi_ctrl del profile --service 11 --profile.name test_nat_4000
fdpi_ctrl load profile --service 11  --profile.name test_nat_4000 --
profile.json '{ "nat_ip_pool" : "111.111.111.0/30", "nat_tcp_max_sessions" :
4000, "nat_udp_max_sessions" : 4000, "nat_type" : 0 }'
fdpi_ctrl load --service 11 --profile.name test_nat_4000 --file
save_users.txt
```

Do not forget to change the pool name and its new parameters in the commands given above according to your actual settings.

How to allocate a specific address to a NAT 1:1 subscriber?

If the subscriber has only one private address and you want to give him a specific public address, you need to take into consideration the dependance between private and public addresses caused by the non-blocking address dispatching algorithm in the DPI.

```
subscriber_public_address & mask = subscriber_private_address & mask
```

here, mask depends on the number of worker threads:

- for 4 worker threads mask=3 (typically for Stingray SG >= 10)
- for 2 worker threads mask=1 (typically for Stingray SG <= 6)

In fact, for newer SSG versions, subscribers with even private addresses need to be given even public addresses, and odd ones - odd ones. It is enough to take into account only the lower byte NNN in the IP-address XXX.YYY.ZZZ.NNN

Accordingly, for older versions, the equality of the 2 least significant bits of the IP address must be taken into account.

With one worker thread, the dependency between addresses disappears.

You can view the exact mask value in the DPI log:

```
grep nat_hash_mask /var/log/dpi/fastdpi_alert.log
```

If the start was a long time ago, then reload

```
service fastdpi reload
```

Thus, this partially deterministic allocation scheme assumes that private addresses will also be issued to the subscriber statically. In cases where it is necessary to issue a specific public IP address (by contract) and the current private address of the subscriber does not match the above formula, then you will need to change the subscriber's private address to the one that corresponds to the formula.

Example for SSG-20: We need to allocate a public address 188.99.99.27 to a subscriber with a private address 10.0.0.15
mask=3
15&3=3 equals 27&3=3 - this means that such an address can be issued (otherwise it would be necessary to change either the private address given to the subscriber, or the public one assigned to him)

Assign the address to the subscriber with the command:

```
fdpi_ctrl load profile --ip 10.0.0.15 --service 11 --profile.json '{
"nat_ip_pool" : "188.99.99.27/32", "nat_type" : 1 }'
```

NAT Diagnostics

1. A profile must have pools of the same size. Correct:

```
type_profile=1, ref_cnt=0       d3      { "nat_ip_pool" :
"1.1.2.0/28,1.1.3.0/28", "nat_tcp_max_sessions" : 2000,
"nat_udp_max_sessions" : 2000, "nat_type" : 0 }            11      (0x400)
```

Incorrect:

```
type_profile=1, ref_cnt=0       d3      { "nat_ip_pool" :
"1.1.2.0/28,1.1.3.0/26", "nat_tcp_max_sessions" : 2000,
"nat_udp_max_sessions" : 2000, "nat_type" : 0 }            11      (0x400
```

2. For blocked subscribers, you should connect different profile, with different pools. Many network devices, when blocked, can generate a large number of requests, which leads to the use of free ports at the public address.

3. Check if the private addresses are evenly distributed over the public addresses in the profile.

```
fdpi_ctrl list all status --service 11 --profile.name nat_pool |grep
whiteip|cut -f7|sort|uniq -c|sort -n
```

4. Check the number of subscribers that use ports more then the $P value. The average subscriber uses about 600 ports.

```
fdpi_ctrl list all status --service 11 --profile.name nat_pool | awk 'BEGIN
{FS="[=| }\t]+"}  $15>$P {print $1, $14, $15}' | wc -l
```

5. Check how addresses are distributed by pools (subnets) in the profile.

```
fdpi_ctrl list all status --service 11 --profile.name nat_pool |grep
whiteip|cut -f7|cut -d"." -f1,2,3|sort|uniq -c|sort -n
```

How to find a subscriber after NAT. Working with abuse letters

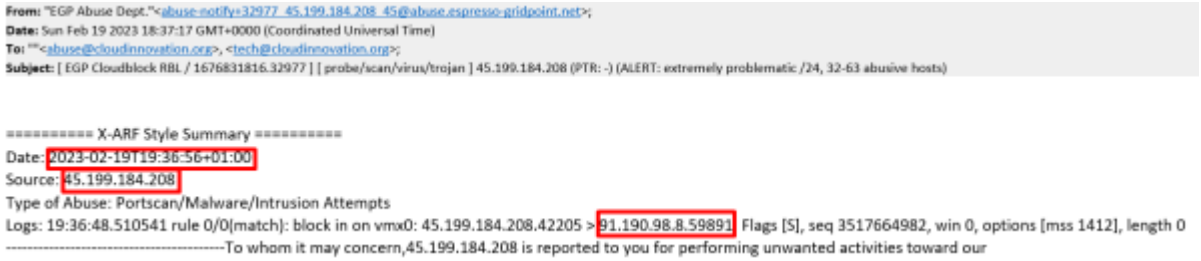
This tutorial is how to find the specific subscriber who is reported abuse. The abuse email usually contains a global address from a NAT pool. We need to understand which of the subscribers went to the resource where the virus activity was detected at a known time behind this NAT-pool. We need to perform **two steps** — find the necessary information in the abuse email and use it to identify the subscriber in the GUI of the Stingray.

Step 1. Research the email

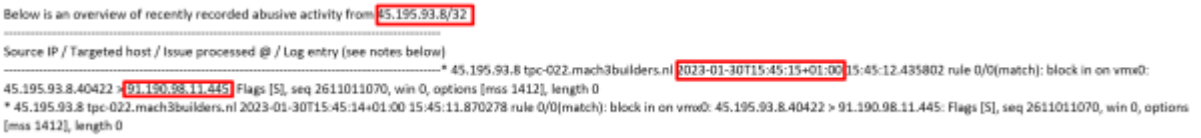
1. The address from your NAT pool (source IP).

2. Address of the attacked resource (destination IP)
3. Activity time on the attacked resource *(considering the time zones!)*

- **Example 1.**

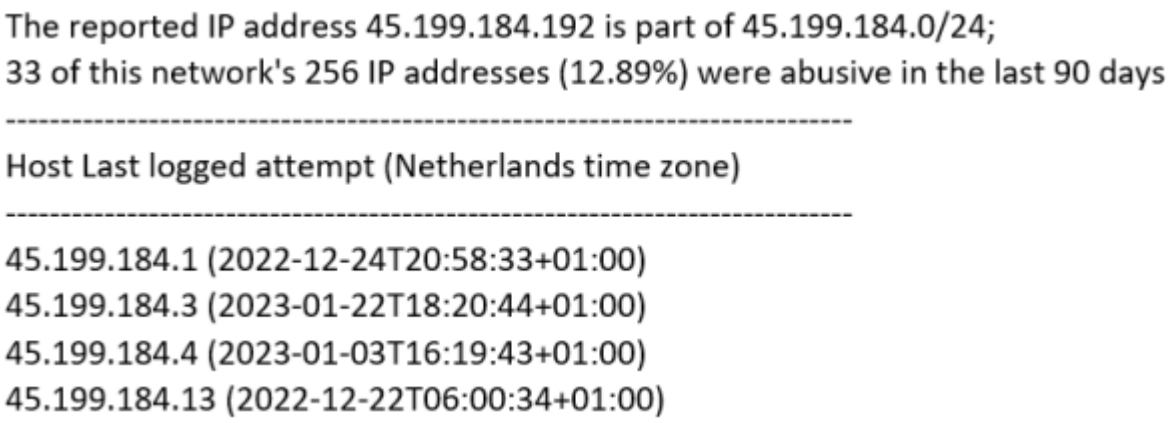


- **Example 2.**



More can be found useful in the email:

1. Reason of abuse
   Date: 2023-02-27T00:53:34+01:00
   Source: 45.199.184.192
   Type of Abuse: Portscan/Malware/Intrusion Attempts
   Logs: 00:53:29.425121 rule 0/0(match): block in on vmx0: 45.199.184.192.65001 > 91.190.98.8.59891: Flags [S], seq 3803861910, win 0, options [mss 1412], length 0
2. History of abuse (if the activity was repeated)



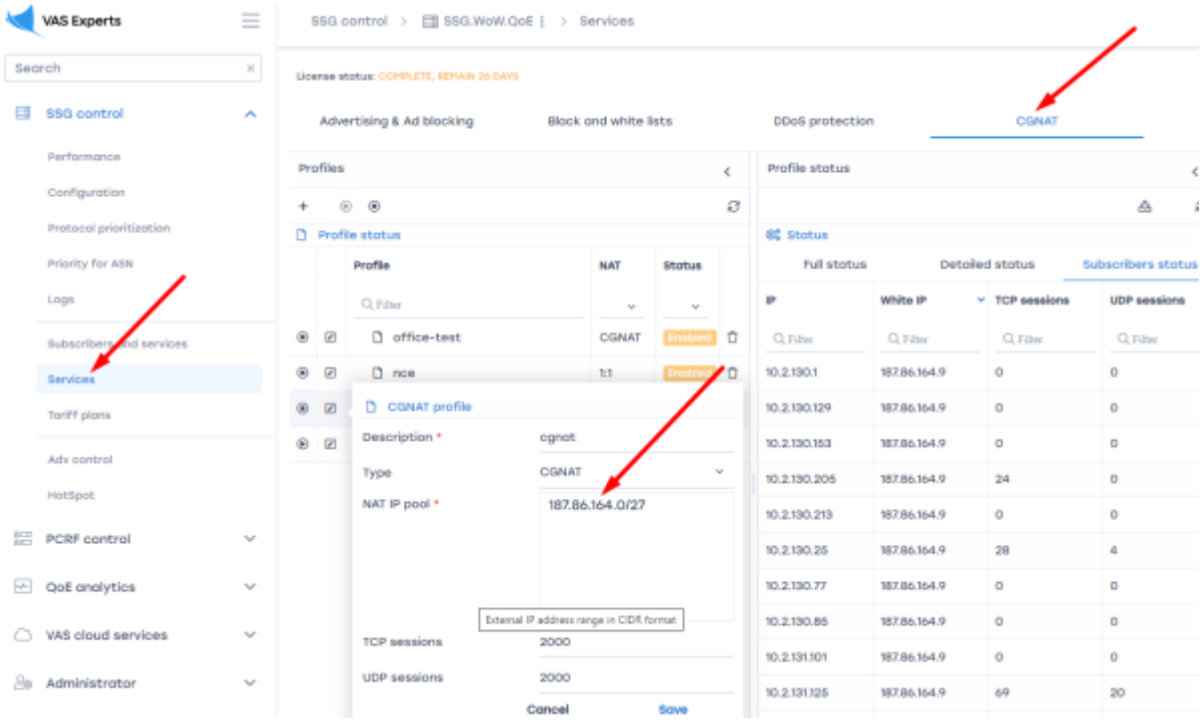
This can help you understand the scope of the problem and identify similar problems on your network.

Step 2. Looking for subscriber activity in the GUI

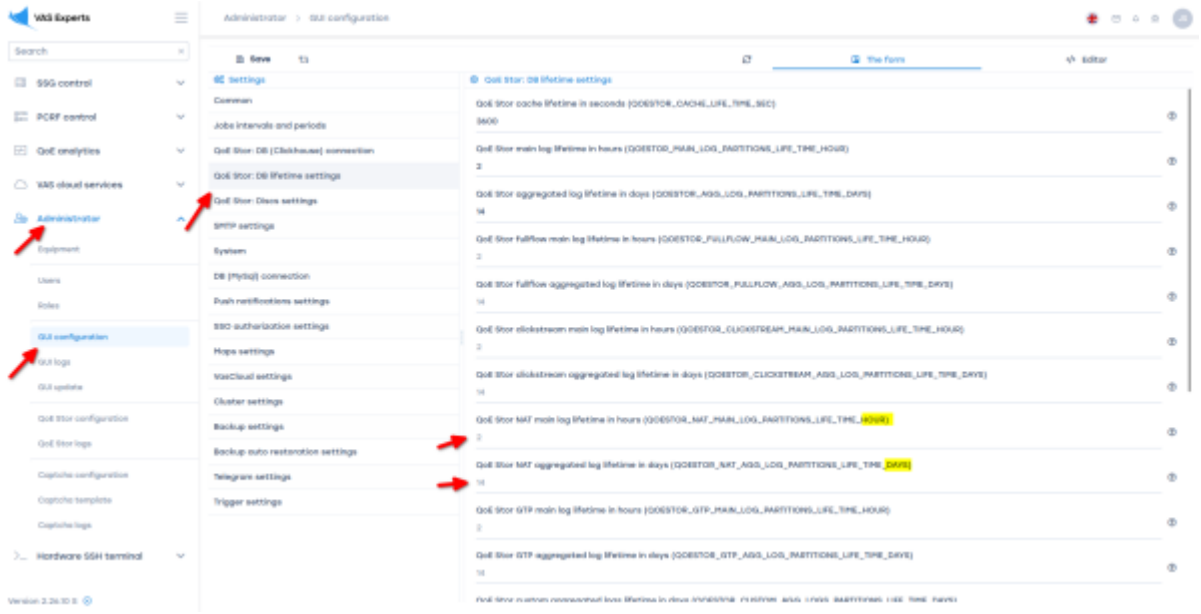
The task is to determine from the logs which subscriber behind the NAT-pool (source IP) specified in the letter was accessing the destination IP at that time.

Before you start the search it is worth checking two facts:

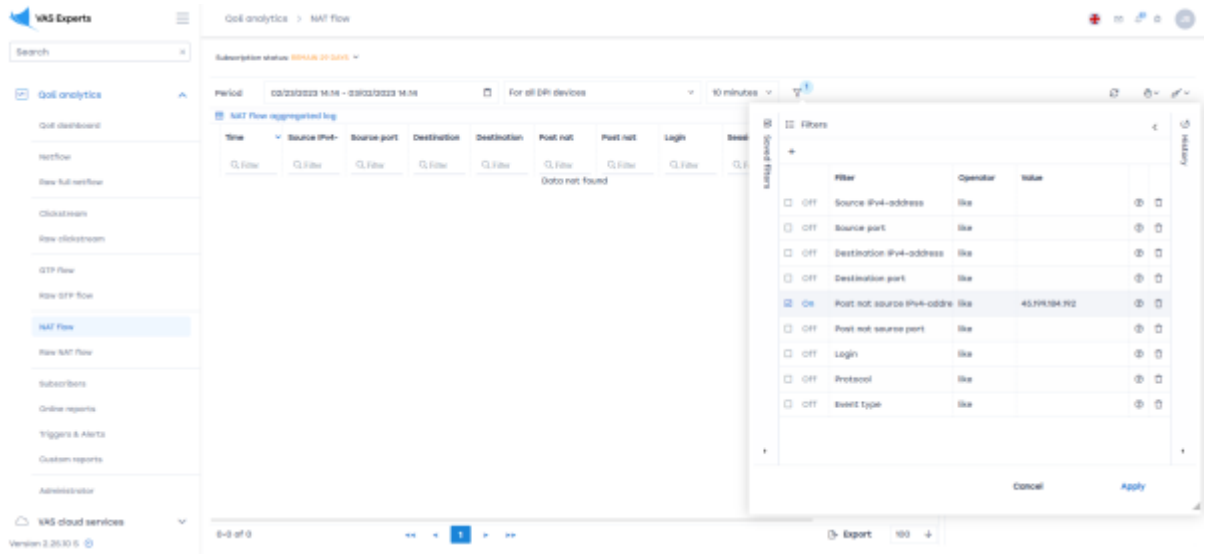
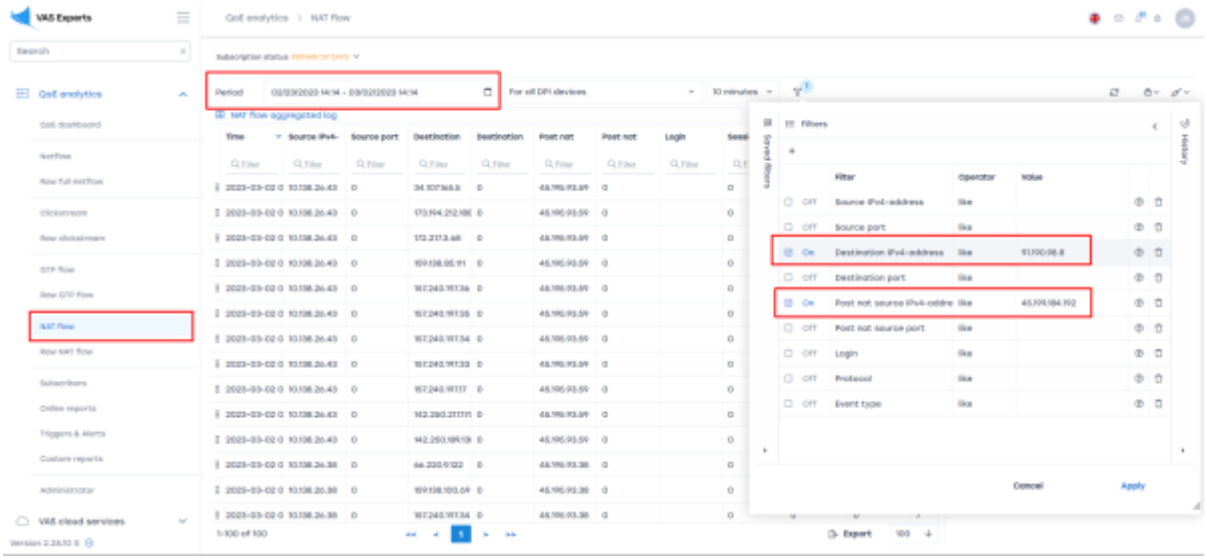
1. The NAT pool in question is set to CG-NAT in Stingray.



2. The NAT log storage time captures the time of activity. View and configure



Then in the GUI you need to open the section NAT flow, select a period, enter the source and destination IP.

- 
- 

Perform the necessary actions with the found subscriber to prevent further abuse.