Содержание

CG-NAT and NAT (Service 11)


Product Description

1. Why is it Recommended to Create a Pool of at Least 2 or 4 Addresses?

The non-blocking dispatch algorithm in DPI, which distributes sessions across worker threads, imposes a limitation on which public IP address can be assigned to a subscriber from the pool:

1. To ensure that a subscriber receives their designated public address, the pool must contain at least as many addresses as there are worker threads (typically 2 for SSG-6 and 4 for SSG-10 and higher). You can determine the number of worker threads with the command

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

2. If the pool contains only one address, it may not be assigned to all subscribers but only to those who fit the load balancing algorithm.

2. How to Determine Which Public Address from the Pool a Subscriber Will Get?

You can see which public address has been assigned to a private one with the command

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

. In NAT 1:1, the public address is allocated immediately upon service assignment; in CG-NAT, it is allocated at the start of the session. The assigned public address is also reported in Radius Accounting for billing logging purposes.

Predicting the exact address that will be assigned to a subscriber from the pool in advance is impossible; it depends on various factors, particularly the current pool load.

3. Inactive SSH Sessions are Closed After Enabling NAT


Parameter descriptions at this link

Indeed, the session lifetime in NAT is limited because the number of sessions per subscriber is a limited resource, and a large number of inactive sessions in the pool reduces NAT and overall performance.

NAT cannot distinguish whether a session became inactive due to an accident or simply because there is no activity, and it closes such long-hanging sessions by inactivity timeout. This behavior is standard and supported by most CG-NAT manufacturers.

In SSG, session lifetimes can be adjusted with the following parameters:

- `lifetime_flow_long=600` — lifetime in seconds of inactive TCP sessions

- `lifetime_flow=60` — lifetime in seconds of other sessions

These parameters should not be set to excessively large values, as the session table may grow too large, impacting CG-NAT performance, and the subscriber may run out of session limits (set in the nat pool parameters).

To maintain long inactive connections, it is recommended to use the `tcp keep-alive` mechanism, where an empty packet is periodically sent in the session, signaling that the session is still active.

You can configure `tcp keep-alive` both individually for the application on the server or client side and at the operating system level for all applications.

Example configuration on the ssh server: add the line to the `/etc/ssh/ssh_config` file:

```
ServerAliveInterval 60
```

Example configuration on the ssh client: add the lines to the `~/.ssh/config` file:

```
Host *
  ServerAliveInterval 60
```

or in the command line:

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

Example configuration for all applications in CentOS: add the lines to the `/etc/sysctl.conf` file:

```
net.ipv4.tcp_keepalive_time = 600
net.ipv4.tcp_keepalive_intvl = 60
net.ipv4.tcp_keepalive_probes = 20
```

4. How Many Private IP Addresses Can Be Hidden Behind One Public IP in CG-NAT?

It is recommended to maintain a ratio from 1:10 (better) to 1:100 (worse).

Details:

By default, 64512 ports (65535-1023, as the first 1024 ports are not used because they are system ports) are available for CG-NAT on one public IP. Each port represents one TCP session and one UDP. The number of sessions created by subscribers varies: individuals create fewer sessions, while corporate users create more (thus, corporate users should use a separate pool with different session limits). A subscriber with torrents can create up to 1000 sessions at peak.

On average, an individual creates 50-60 simultaneous sessions, meaning 64512/60=1075 individuals can be hidden behind one private IP. However, such significant oversubscription is not recommended in practice, as many popular services (mail, video, search) use IP address-based botnet attack protection. If too many requests come from one address, they may be considered an attack and block some requests or enable captcha, causing inconvenience to subscribers.

Also, consider the feature of port release mechanism in the NAT Pool:

1. When Service 11 is enabled for a subscriber, a Public IP is assigned based on the distribution algorithm
2. When a subscriber starts establishing sessions, ports are taken from the common SSG DPI queue and assigned with certain timeouts
3. If a specific Public IP has many subscribers competing for free ports, subscribers may experience access issues.

Recommendations for creating and operating NAT Pools:

1. Subscribers under blocking (Service 5 + policing) should be placed in a separate NAT Pool to avoid impacting active subscribers. For instance, an iPhone may establish many sessions searching for an active service.
2. Create sparse pools and separate clients into different NAT Pools by type: individuals and corporate users.
3. Monitor clients generating high load and work with them. For receiving, processing, and storing NetFlow from DPI, we suggest using the QoE Store software product for statistics collection and the DPIUI2 graphical interface. You can analyze subscriber traffic and conclude that their PC is infected.

5. How to Change the Parameters of an Existing and Used Pool?

Parameter descriptions at this link

1. Changing the session limit:

```
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 }'
```

   Use the pool creation command identical to the previous one but with different `nat_tcp_max_sessions` and `nat_udp_max_sessions` settings.

2. Adding 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 }'
```

   Use the pool creation command identical to the previous one but with an additional pool specified by a comma.

3. Reducing the pool

The current version does not support dynamic pool size reduction and address exclusion. In this case, you will need to free the pool, delete it, and create it with new parameters.

For convenience, install jq (a utility for working with JSON data):

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

Then save the information about the subscribers of the current pool, delete and recreate the pool, and reconnect the subscribers:

```
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 to the ones you need.

6. How to Assign a Specific Address to a Subscriber with NAT 1:1?

The method described below is inactive when `rx_dispatcher=1`

If a subscriber has only one private address and it is necessary to assign a specific public address to the subscriber, the dependency between the private and public addresses must be considered, which is imposed by the non-blocking address dispatching algorithm in DPI.

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

where the mask depends on the number of working threads:

- with 4 working threads, mask=3 (typical for SSG >= 10)
- with 2 working threads, mask=1 (typical for SSG ⇐ 6)

In fact, for older versions of SSG, subscribers with even private addresses should be assigned even public addresses, and subscribers with odd private addresses should be assigned odd public addresses. Only the last byte NNN in the IP address XXX.YYY.ZZZ.NNN needs to be considered.

Accordingly, for newer versions, the equality of the 2 least significant bits of the IP address should be considered.

With one working thread, the dependency between the addresses disappears.

The exact mask value can be found in the DPI log:

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

If the service has been running for a long time, perform a reload:

```
service fastdpi reload
```

Thus, this partially deterministic distribution scheme essentially assumes that private addresses will also be statically assigned to the subscriber. In cases where a specific public IP address is specified in the contract and the current private address of the subscriber does not fit the formula mentioned above, it will be necessary to change the subscriber's private address to one that matches the formula.

Example for SSG-20: A subscriber with a private address of 10.0.0.15 needs to be assigned a public address of 188.99.99.27
mask=3
15&3=3 equals 27&3=3 - this means that the address can be assigned (otherwise, either the private address assigned to the subscriber or the public address intended for the subscriber would need to be changed).

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 }'
```

Parameter description via this link

7. NAT Diagnostics

Parameter description via this link

1. The pools in the profile must be of the same size [1)]. 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 subscribers who are blocked, a different profile with different pools should be applied. Many network devices, when blocked, can generate a large number of requests, leading to the consumption of free ports on the public address.

3. Check the uniformity of the distribution of private addresses across 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 using ports beyond the value of the $P variable. On average, a 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 the addresses are distributed across 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
```

1)

The requirement is not relevant if `rx_dispatcher=1` or `rx_dispatcher=2`