

## Access Rights Policy based on IP Range

- **Product:** Rosetta
- **Product Version:** 8.1 and newer

### Question

How Rosetta checks the IP during delivery using Access Rights Policy based on "IP Range" expression/s ?

### Answer

Rosetta until version 8.1 only supports IPv4 in Access Rights Policy "IP Range" field.

Initial IPv6 support was added in Rosetta version 8.2: either IPv4 or IPv6 "IP Range" expressions are supported within the same Access Right Policy.

Rosetta version 8.3 is tentatively planned to have support for combination of IPv4 and IPv6 "IP Range" expressions in the same Access Right Policy.

During entity (IE/REP/FILE) delivery, the relevant Access Rights Policy is checked.

When the Access Rights Policy is based on one or more "IP Range" expressions, Rosetta first checks if the HTTP request contains a header, as configured in "client\_ip\_header" General Parameter of Rosetta (typically "X-Forwarded-For").

#### Until Rosetta version 8.1:

If the "client\_ip\_header" HTTP header (i.e. "X-Forwarded-For") value exists in the delivery request, Rosetta checks if the header's value is a valid IPv4 value (If multiple such headers exist, Rosetta checks the value of such first header).

Then:

- If it's a valid IPv4 value, Rosetta compares it with the Access Rights Policy conditions. ([More details on how Access Rights work](#))
- If it's not a valid IPv4 value, (like when IPv6 is used, or that the "client\_ip\_header" header's value is a string of comma delimited IPs, etc), Rosetta uses the REMOTE ADDRESS: (i.e. the first left column in Tomcat access log)
  - If it's a valid IPv4 value - Rosetta compares it with the Access Rights Policy conditions. ([More details on how Access Rights work](#))
  - If it not a valid IPv4 value - Rosetta declines the request (assuming any other Access Rights Policy expression outcome is false as well)

#### **Example 1**

```
192.168.0.1 "192.168.0.2" - - [22/Aug/2023:01:46:29 +0200] GET /delivery/...
```

Assuming the "client\_ip\_header" is set to "X-Forwarded-For", Rosetta would compare the relevant Access Rights Policy conditions against "192.168.0.2".

#### **Example 2**

```
192.168.0.1 "0:0:0:0:0:0:2" - - [22/Aug/2023:01:46:29 +0200] GET /delivery/...
```

Assuming the "client\_ip\_header" is set to "X-Forwarded-For", Rosetta would ignore the non valid IPv4 value and would compare the relevant Access Rights Policy conditions against the remote address value 192.168.0.1.

### Example 3

```
0:0:0:0:0:0:2 "-" - - [22/Aug/2023:01:46:29 +0200] GET /delivery/...
```

Assuming the "client\_ip\_header" is set to "X-Forwarded-For" and no value is passed, Rosetta would compare the relevant Access Rights Policy conditions against the remote address "0:0:0:0:0:0:2", but since this value is not IPv4 the request would get rejected since Access Rights "IP Range" field support only IPv4.

### **Rosetta version 8.2 and newer:**

If the "client\_ip\_header" HTTP header (i.e. "X-Forwarded-For") value exist in the delivery request, Rosetta checks if the header's value is a valid IPv4 or IPv6 value (If multiple such headers exist, Rosetta checks the value of such first header). Then:

- If it's a valid IPv4 or IPv6 value, Rosetta compares it with the Access Rights Policy conditions. ([More details on how Access Rights work](#))

---

#### Note

Rosetta v8.2.0.x support the same IP version across any "IP Range" expressions of the same Access Rights Policy - i.e. combining IPv4 and IPv6 "IP Range" expressions (in the same or different Group) in the same Access Right Policy is not supported.

Rosetta v8.3 is tentatively planned to have support for combination of IPv4 and IPv6 "IP Range" expressions in the same Access Right Policy.

- 
- If it's not a valid IPv4 or IPv6 value, (like when unexpected characters are added by a proxy, etc), then Rosetta checks the HTTP Remote Address (i.e. the first left column as reported by Tomcat access log):
    - If it's a valid IPv4 or IPv6 value - Rosetta compares it with the Access Rights Policy conditions. ([More details on how Access Rights work](#))
    - If it not a valid value - Rosetta declines the request (assuming any other Access Rights Policy expression outcome is false as well)

### Example 4

```
192.168.0.1 "-" - - [22/Aug/2023:01:46:29 +0200] GET /delivery/...
```

or

```
0:0:0:0:0:0:1 "-" - - [22/Aug/2023:01:46:29 +0200] GET /delivery/...
```

Assuming the "client\_ip\_header" is set to "X-Forwarded-For" but since its value is empty, Rosetta compares the relevant Access Rights Policy conditions against the HTTP

Remote Address value (e.g. 192.168.0.1 or 0:0:0:0:0:0:0:1).

#### Example 5

```
192.168.0.1 "0:0:0:0:0:0:5" -- [22/Aug/2023:01:46:29 +0200] GET /delivery/...
```

or

```
192.168.0.1 "10.0.0.1" -- [22/Aug/2023:01:46:29 +0200] GET /delivery/...
```

Assuming the "client\_ip\_header" is set to "X-Forwarded-For" and its value is a valid IPv4 (e.g. 10.0.0.1) or IPv6 (e.g. 0:0:0:0:0:0:0:5), Rosetta compares the "X-Forwarded-For" value against the relevant Access Rights Policy.

#### Example 6

```
192.168.0.1 "10.0.0.1," -- [22/Aug/2023:01:46:29 +0200] GET /delivery/...
```

or

```
192.168.0.1 "0:0:0:0:0:0:6," -- [22/Aug/2023:01:46:29 +0200] GET /delivery/...
```

Assuming the "client\_ip\_header" is set to "X-Forwarded-For" and its value is not a valid IPv4 or IPv6 (e.g. due to an unexpected postfix of the comma character) Rosetta checks the HTTP Remote Address (e.g. 192.168.0.1) against the relevant Access Rights Policy.

#### Note

Tomcat, bundled out-of-the-box with Rosetta, may identically log multiple "X-Forwarded-For" HTTP headers and also single "X-Forwarded-For" header containing a delimited multi-value string.

For example, in the scenario "X-Forwarded-For" header is either:

A. Two different headers with different values like:

- 0:0:0:0:0:0:1
- 0:0:0:0:0:0:2

B. Single header with comma and space delimited multi-values: 0:0:0:0:0:0:1, 0:0:0:0:0:0:2

In both cases, tomcats may log such request identical in the access log, e.g.:

```
192.168.0.1 "0:0:0:0:0:0:1, 0:0:0:0:0:0:2" -- [22/Aug/2023:01:46:29 +0200] GET /delivery/...
```

However, Rosetta treats scenario B value (i.e. a delimited string "X-Forwarded-For" value) as an invalid IP value.

#### Note

Rosetta version 8.2 and newer versions support the following IPv6 syntax:

- Expanded address (eight colon delimited groups, each comprised of four hexadecimal digits)

---

For example:

█ 2620:012a:8001:0000:0000:0000:0000:0004

- Omitted leading zeros

For example:

█ 2620:12a:8001:0:0:0:0:4

- Compressed fields (using "::")

For example:

█ 2620:12a:8001::4

- IPv4-mapped IPv6

For example:

█ ::ffff:23.185.0.4

**NOTE:** Square brackets are not supported. These characters should be filtered out by an external network component (e.g. Proxy, Load-Balancer, etc).

---

---

- **Article last edited:** 30-Apr-2025