Breaking Internet Routing (outages that didn’t happen)

Job Snijders

Fastly

job@fastly.com

CERN Academic Training
March 2022

Job Snijders

Internet Routing System Hacker

Principal Engineer @ Fastly

Developer @ OpenBSD

Board member @ RIPE NCC

Board member @ PeeringDB

Board member @ RSSF

Co-chair IETF GROW WG

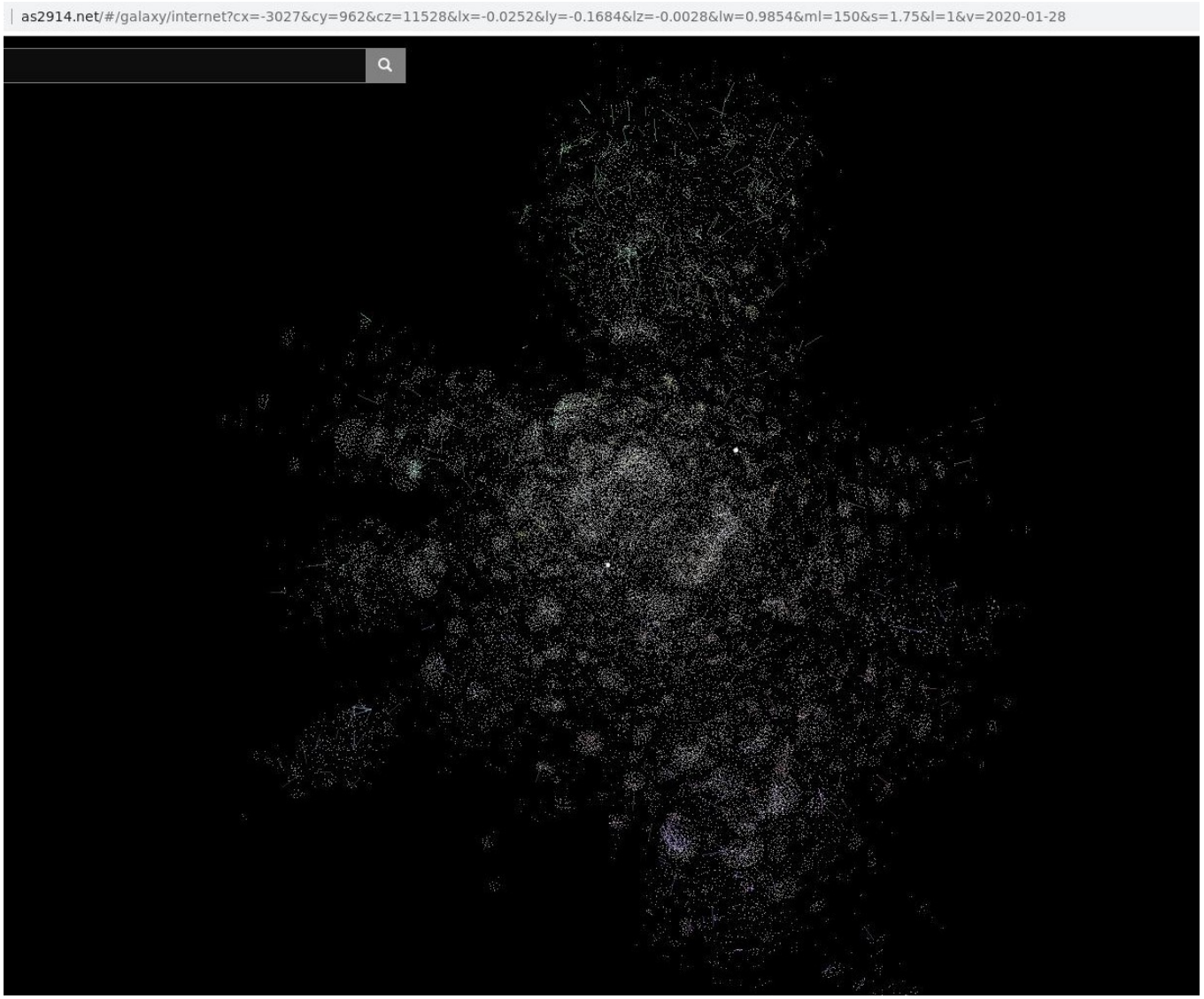
Co-chair RIPE Routing-WG

https://twitter.com/JobSnijders

Agenda

- **Introduction**
- **How does the global Internet routing system work?**
  - Designed to grow
  - BGP
- **Attempts to curb the unwieldy**
  - IRR
  - RPKI
- **How does RPKI work?**
  - Imposing constraints on the chaos using a PKI
  - Technical details
  - RPKI-To-Router protocol (RTR)
- **Known negative interactions between RPKI and BGP**
  - Router bugs
  - Inefficient Routing Policies: grouping RPKI Validation States as BGP Communities considered Harmful
- **Wrapping it up!**
- **Questions?**

How does the Internet routing system work?



http://as2914.net/

RFC 4271

How to curb such an unwieldy system?

EBGP Routers tell each other what you can reach through them.

Auxiliary systems need to exist to impose *order*.

Order means routers know what IP prefix belongs with what AS

Two systems: first IRR (in ancient times), and now RPKI!

How to curb such an unwieldy system?

IRR is a plain-text system:

```
$ whois -h whois.ripe.net 2001:67c:208c::


route6:          2001:67c:208c::/48
descr:           NL-SNIJDERS-IT
origin:          AS15562
mnt-by:          SNIJDERS-MNT
created:         2015-08-31T14:16:27Z
last-modified:   2015-08-31T14:16:27Z
source:          RIPE
```

How to curb such an unwieldy system?

Downsides of Internet Routing Registry (IRR) system:

- No transport security (port 43!)
- No object security (gotta hope for the best at face value)
- No way of verifying the source's authentication process

How to curb such an unwieldy system?

Downsides of Internet Routing Registry (IRR) system:

- No transport security (port 43!)
- No object security (gotta hope for
- No way of verifying the source's a

How does RPKI work?

Overall architecture: RFC 6480

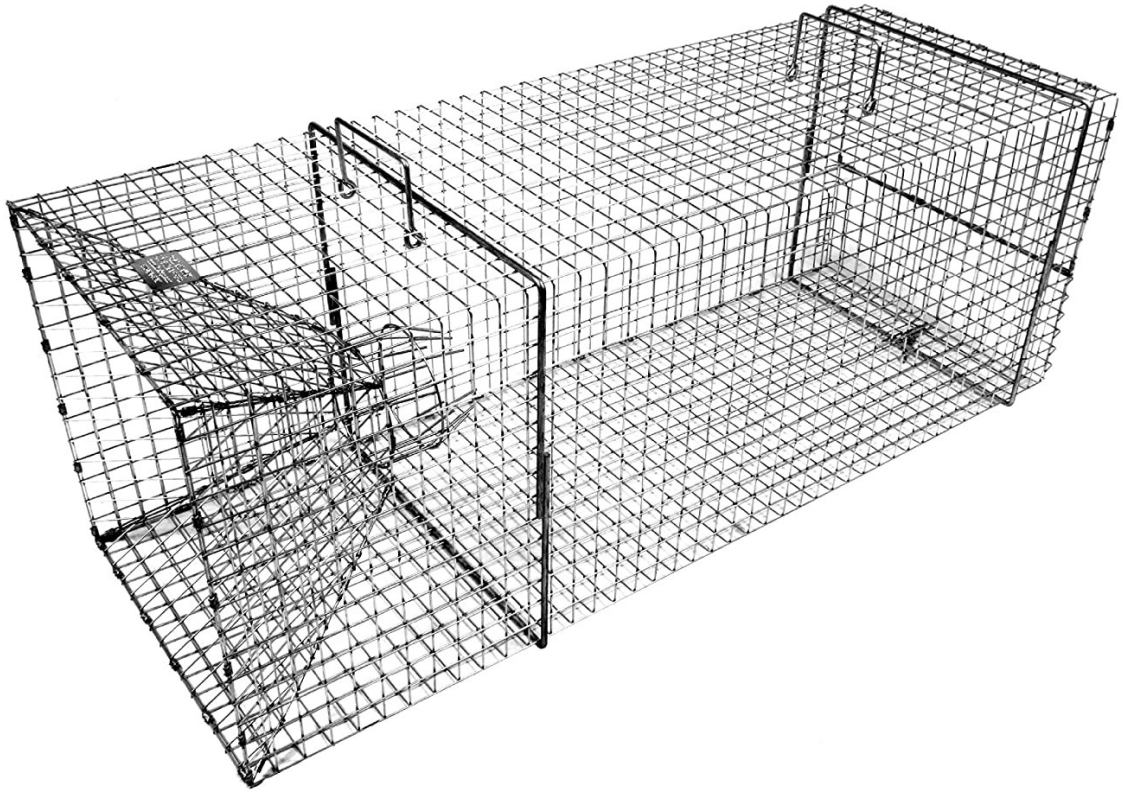
RPKI has *Object Security*!

A tree of X.509 certificates following address allocation

At the leaves of the tree *Route Origin Authorizations* exist

How does RPKI work? Every delegation is a “trap”

In the chain of certificates: You are *constrained* by the list of *subordinate* resources on your parent



How does RPKI work? A Trust Anchor Certificate

```
RPKI Root Certificate
Location:                    ta/ripe/ripe-ncc-ta.cer
Subject key identifier:
                E8:55:2B:1F:D6:D1:A4:F7:E4:04:C6:D8:E5:68:0D:1E:BC:16:3F:C3

Authority key identifier: Trust Anchor
Manifest:                    rpki.ripe.net/repository/ripe-ncc-ta.mft
CA Repository:               rsync://rpki.ripe.net/repository/
Notification URL:            https://rrdp.ripe.net/notification.xml

Subordinate resources:
    until: 2117-11-28T14:39:55Z
    Resources:
    1: AS: 0 – 4294967295
    1: IP: 0.0.0.0/0
    2: IP: ::/0
```

https://console.rpki-client.org/ta/ripe/ripe-ncc-ta.cer.html

A RPKI intermediate cert operated by RIPE NCC

```
RPKI Certificate
Location:                 rpki.ripe.net/repository/aca/KpSo3VVK5wEHIJnHC2QHVV3d5mk.cer

Subject key identifier:   2A:94:A8:DD:55:4A:E7:01:07:20:99:C7:0B:64:07:55:5D:DD:E6:69
Authority key identifier: 2A:7D:D1:D7:87:D7:93:E4:C8:AF:56:E1:97:D4:EE:D9:2A:F6:BA:13
Manifest:                 rpki.ripe.net/repository/DEFAULT/KpSo3VVK5wEHIJnHC2QHVV3d5mk.mft
CA Repository:            rpki.ripe.net/repository/DEFAULT/
Notification URL:         https://rrdp.ripe.net/notification.xml

Subordinate resources:
    until: 2023-07-01T00:00:00Z
    Resources:
    1: AS: 7
    2: AS: 28
    3: AS: 137
    4: AS: 224
    5: AS: 248 -- 251
    6: AS: 261
    … etc etc …
```

https://console.rpki-client.org/rpki.ripe.net/repository/aca/KpSo3VVK5wEHIJnHC2QHVV3d5mk.cer.html

A RPKI CA certificate belonging to a RIPE member

```
RPKI Certificate

Location:                  rpki.ripe.net/repository/DEFAULT/OOFPkv3HzPv8GCNhUjrifWl-lS8.cer

Subject key identifier:    38:E1:4F:92:FD:C7:CC:FB:FC:18:23:61:52:3A:E2:7D:69:7E:95:2F
Authority key identifier:  2A:94:A8:DD:55:4A:E7:01:07:20:99:C7:0B:64:07:55:5D:DD:E6:69
Manifest:                  chloe.sobornost.net/rpki/RIPE-nljobsnijders/OOFPkv3HzPv8GCNhUjrifWl-
lS8.mft
CA Repository:             chloe.sobornost.net/rpki/RIPE-nljobsnijders/
Notification URL:          https://chloe.sobornost.net/rpki/news.xml

Subordinate resources:

    until: 2023-07-01T00:00:00Z

    Resources:
    1: AS: 15562
    1: IP: 45.138.228.0/22
    2: IP: 2001:67c:208c::/48
    3: IP: 2a0e:b240::/29
```

https://console.rpki-client.org/rpki.ripe.net/repository/DEFAULT/OOFPkv3HzPv8GCNhUjrifWl-lS8.cer.html

Zooming in on a ROA at a leaf of the RPKI

Route Origin Authorization

```
Location:                chloe.sobornost.net/rpki/RIPE-nljobsnijders/8EjgZ6BLB_EFHp9nPxEgX5icjjM.roa

Subject key identifier:   F0:48:E0:67:A0:4B:07:F1:05:1E:9F:67:3F:11:20:5F:98:9C:8E:33
Authority key identifier: 38:E1:4F:92:FD:C7:CC:FB:FC:18:23:61:52:3A:E2:7D:69:7E:95:2F

asID:                    AS15562

Prefixes:

    1: 2001:67c:208c::/48 maxlen: 48
    2: 2a0e:b240::/48 maxlen: 48
```

https://console.rpki-client.org/chloe.sobornost.net/rpki/RIPE-nljobsnijders/8EjgZ6BLB_EFHp9nPxEgX5icjjM.roa.html

Inspecting a RPKI ROA with OpenBSD rpki-client

```
$ firefox https://repology.org/project/rpki-client

$ sudo apt install rpki-client && sudo rpki-client       # now make coffee

$ rpki-client \
    -t /etc/rpki/ripe.tal \
    -f rsync://chloe.sobornost.net/rpki/RIPE-nljobsnijders/8EjgZ6BLB_EFHp9nPxEgX5icjjM.roa


File: chloe.sobornost.net/rpki/RIPE-nljobsnijders/8EjgZ6BLB_EFHp9nPxEgX5icjjM.roa
Subject key identifier: F0:48:E0:67:A0:4B:07:F1:05:1E:9F:67:3F:11:20:5F:98:9C:8E:33
Authority key identifier: 38:E1:4F:92:FD:C7:CC:FB:FC:18:23:61:52:3A:E2:7D:69:7E:95:2F
Authority info access: rsync://rpki.ripe.net/repository/DEFAULT/OOFPkv3HzPv8GCNhUjrifWl-lS8.cer
ROA valid until: Jul 01 00:00:00 2022 GMT
AsID: 15562
    1: 2001:67c:208c::/48 maxlen: 48
    2: 2a0e:b240::/48 maxlen: 48
Validation: OK
```

How does RPKI apply to BGP?

BGP Prefix Origin Validation: RFC 6811

How does RPKI apply to BGP?

BGP Prefix Origin Validation: RFC 6811

```
$ fgrep 193.0.0.0/21 /var/db/rpki-client/openbgpd
    193.0.0.0/21 source-as 3333 expires 1648622587
```

```
$ bgpctl show rib 193.0.6.139
flags: * = Valid, > = Selected, I = via IBGP, A = Announced,
     S = Stale, E = Error
origin validation state: N = not-found, V = valid, ! = invalid
origin: i = IGP, e = EGP, ? = Incomplete

flags ovs destination      gateway       lpref  med aspath origin
I     V 193.0.0.0/21       165.254.255.1 100  1000 2914 12859 3333 i
```

How does RPKI apply to BGP?

BGP Prefix Origin Validation: RFC 6811

The validation algorithm has 3 possible outcomes:

Valid: A ROA exists, and the BGP route conforms to the ROA

Invalid: covering ROAs exist, but none of them permits the route

Not-Found: no covering ROA exists for the BGP route

How does RPKI apply to BGP?

BGP Prefix Origin Validation: RFC 6811

The validation algorithm has 3 possible outcomes:

Valid: A ROA exists, and the BGP route conforms to the ROA

Invalid: cover

THIS STATE IS WHAT NETWORK OPERATORS REJECT

Not-Found: no covering ROA exists for the BGP route

Contextual input validation – IMPORTANT! :-)

A RPKI ROA has the following datastructure layout:

```
RouteOriginAttestation: {
  - asID: INTEGER
  - ipAddrBlocks: {
  - ROAIPAddressFamily: {
  - addressFamily
    - ROAIPAddress: {
      - address: BIT STRING
      - maxLength: INTEGER }
    }
  }
}
```

Contextual input validation – IMPORTANT! :-)

MaxLength is a silly ‘macro’, here is an example

2001:67c:208c::/48 maxlength 50

Equals

2001:67c:208c::/48 + 2001:67c:208c::/49
2001:67c:208c:8000::/49 + 2001:67c:208c::/50 +
2001:67c:208c:c000::/50 + 2001:67c:208c:8000::/50 +
2001:67c:208c:bfff::/50 + 2001:67c:208c:c000::/50
….. maxlength is just shorthand notation!

Contextual input validation – IMPORTANT! :-)

Zooming in on an 'openssl asn1parse' of the 2001:db8::/32 entry, it is possible to legally encode 129!!! (hex: 81)

```
33:d=3  hl=2 l=  13 cons:    SEQUENCE
35:d=4  hl=2 l=  11 cons:     SEQUENCE
37:d=5  hl=2 l=   5 prim:      BIT STRING
   0000 - 00 20 01 0d b8
. ...
44:d=5  hl=2 l=   2 prim:      INTEGER           :81
```

Contextual input validation – IMPORTANT! :-)

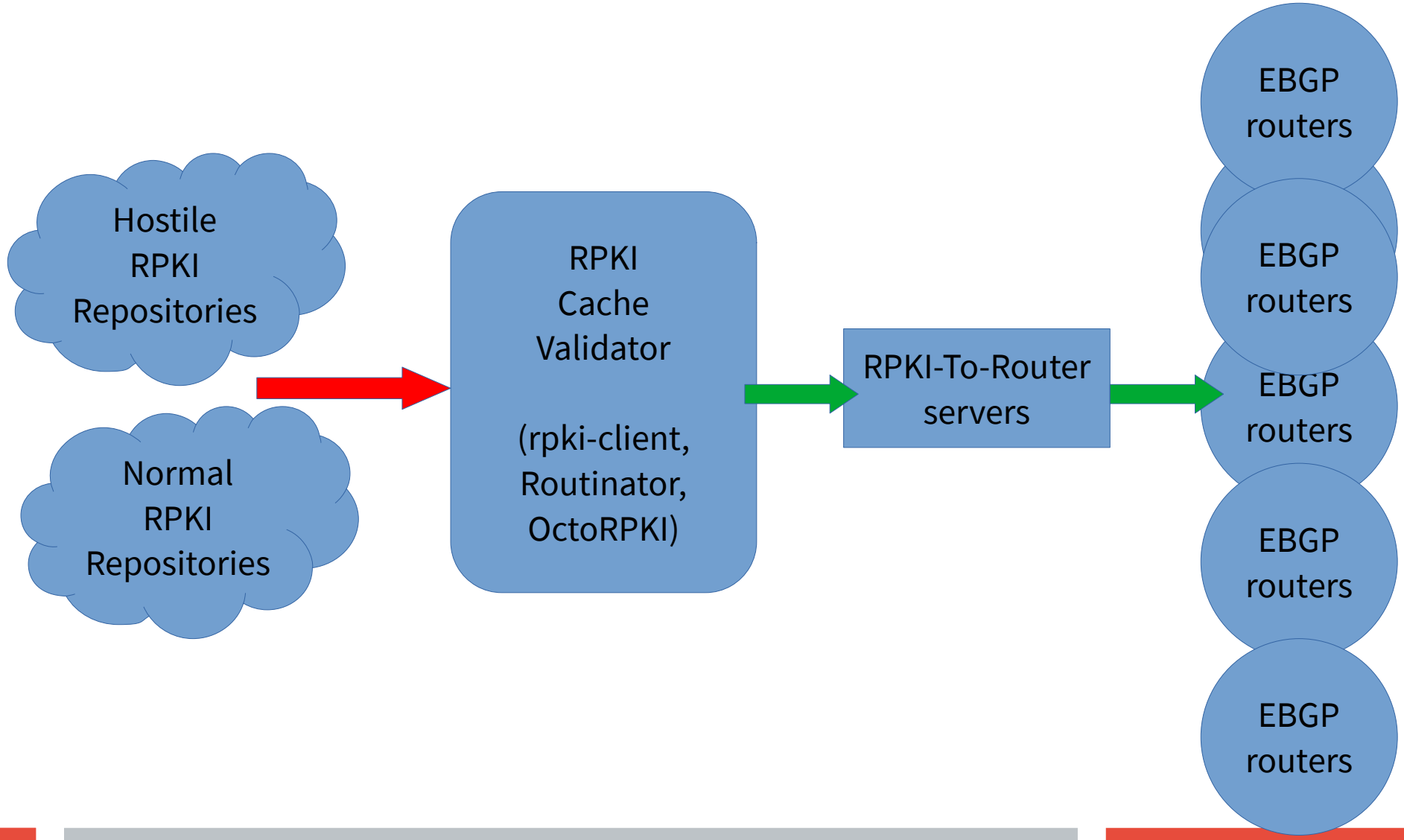
MaxLength doesn’t mean anything in this case:

2001:67c:208c::/48 maxlength 129

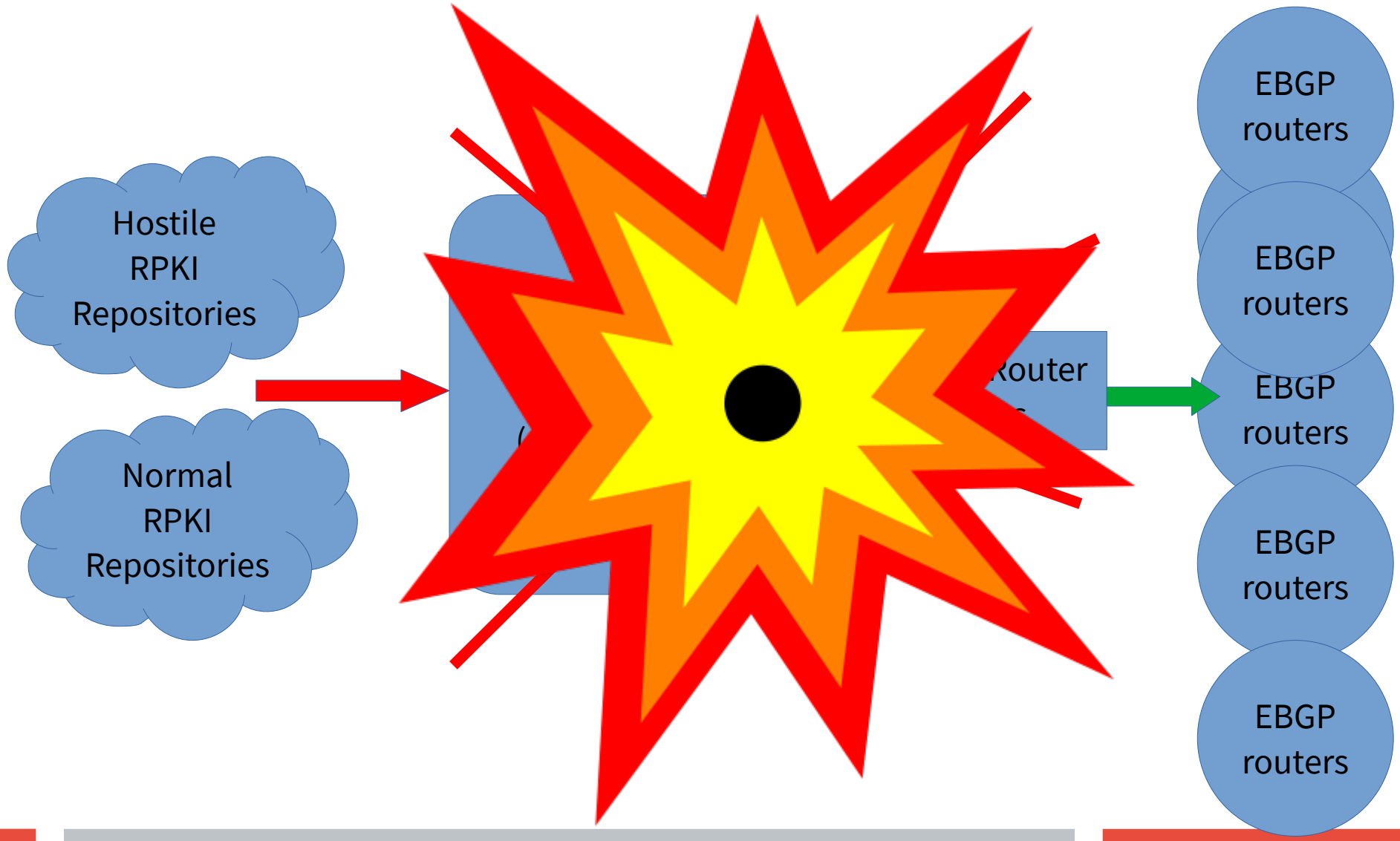
Equals

2001:67c:208c::/48… eh ??? ???? ?? ???????

Flow of information: validators are “firewalls”



Contextually overflowing maxlen: KABLOOEY!



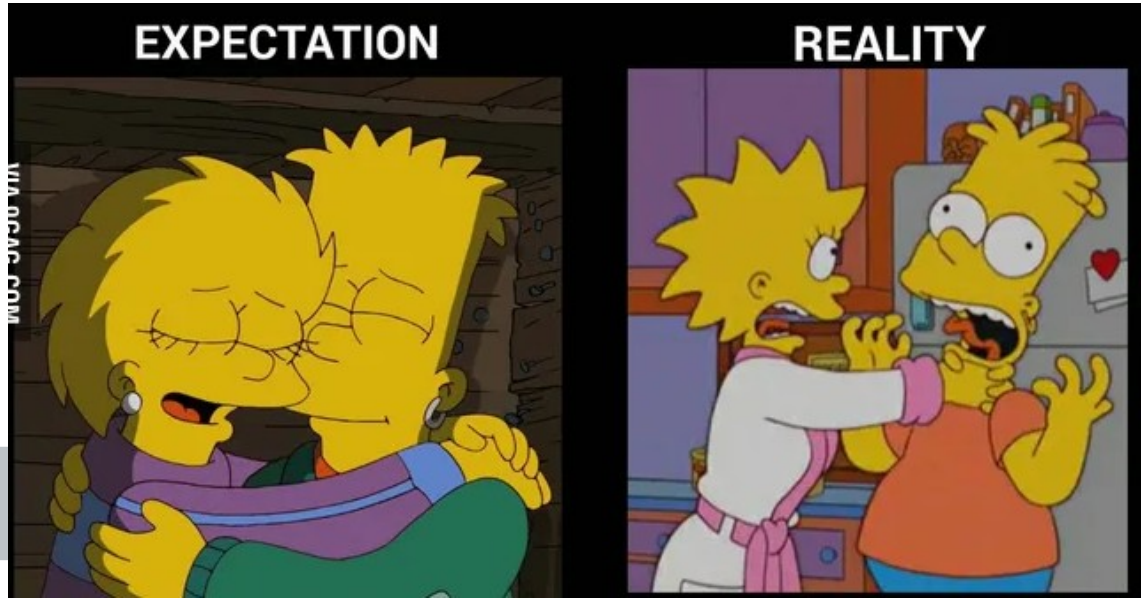
Not a disaster when RTR sessions go down… right?

Expectation: all Valid BGP routes flip to Not-Found

Reality in the year 2020/2021 *(now fixed)*:

- Some ISPs modify *BGP Communities* based on ROV Validation state. Potential result: *33% of the BGP default-free zone churns*
- Some BGP Implementations have bugs: *briefly flip to invalid*

The two above aspects influence each other!



Wrapping it all up!

Folks, always check your inputs!

Identifiers: CVE-2021-3761, CVE-2021-41531, etc

Full write-up:

http://sobornost.net/~job/invalid-maxlength-triggers-rtr-session-termination.txt

Practical RPKI ROV Deployment guidelines

It is considered harmful to manipulate BGP Path Attributes (for example LOCAL_PREF or COMMUNITY) based on the RPKI Origin Validation state.

Making BGP Path Attributes dependent on RPKI Validation states introduces needless brittleness in the global routing system as explained here. Additionally, the use of RFC 8097 is STRONGLY ABSOLUTELY NOT RECOMMENDED. RFC 8097 has caused issues for multi-vendor network operators.

https://bgpfilterguide.nlnog.net/guides/reject_invalids/

Questions?

Discussion?

Comments?

job@fastly.com