RIPE NCC
RIPE NETWORK COORDINATION CENTRE

# KISS your AS PATH

## RIPE RIS analysis for outdated technologies and anomalies

**Alex Semenyaka | RIPE NCC |** 30 September 2026

# Introduction

# RIPE RIS

## What is it?

- RIS is a routing data collection platform: it collects **events on BGP**, the protocol by which traffic is routed between networks on the Internet
- RIS employs a globally distributed set of Remote Route Collectors
- **RIS performs (almost) passive observations on the control plane**



# BGP in the Internet

## A short recap

Route information is transferred in BGP via **UPDATE messages**

UPDATE contains **ANNOUNCEMENTS** (new routes)
and WITHDRAWALS (routes disappearance)

UPDATE contains the route itself (**NLRI**) and its **attributes**

One of the mandatory attributes is **AS_PATH**

*(The autonomous systems traversed by UPDATE from the route originator)*

# AS_PATH attribute

## Two roles

- The AS_PATH length is used in the **BGP Path Selection Process**
  - the shorter, the better

- The AS_PATH is used to **prevent loops**
  - we do not accept an announcement if our own autonomous system is already in the AS_PATH
    - There are some exceptions

# Example (simplified)

## A random UPDATE from RIPE RIS Live

**AS_PATH value**

Originator of the route is AS199340,
and 5 ASes were passed to achieve us

**NLRI value**

Prefix is `2a06:de05:6065::/48`


```
// Received at 16:08:38 (92.62 second delay)
{
    "timestamp": 1790514426.04,
    "peer": "198.58.198.252",
    "peer_asn": "1403",
    "id": "198.58.198.252-01a0e2f9f0b80009",
    "host": "rrc00.ripe.net",
    "type": "UPDATE",
    "path": [1403, 577, 1299, 174, 53667, 199340],
    "community": [[1403, 140]],
    "origin": "IGP",
    "announcements": [
        {
            "next_hop": "198.58.198.252",
            "prefixes": [
                "2a06:de05:6065::/48"
            ]
        }
    ],
    "withdrawals": [],
    "raw":
"FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF005F020000004840010100400021A02060000057B00000241000
00513000000AE0000D1A300030AACC00804057B008C900E001C000201100000000000000000000FFFFC
63AC6FC00302A06DE056065"
}
```

# AS_PATH attribute

## Diving deeper into the structure

In general, AS_PATH is not a simple list of ASNs

The BGP-4 specification (RFC 4271) defines it as a **sequence of segments**

There are **four types** of segments

(**AS_CONFED* are prohibited in the public Internet**)

| Notation in this slide deck | RFC 4271 segment type | Meaning |
|---|---|---|
| 1 2 3 | AS_SEQUENCE | an ordered list (the one everyone knows) |
| {1,2,3} | AS_SET | an unordered set of ASes |
| (1 2 3) | AS_CONFED_SEQUENCE | an ordered list within the confederation |
| [1,2,3] | AS_CONFED_SET | an unordered set of ASes within the confederation |

# AS_PATH segments

## Why do we need sets?

**AS_SEQUENCE** segments are straightforward and familiar

**AS_CONFED_SEQUENCE** segments appear in the internal infrastructure of a large operator when it uses BGP confederations

The less common method of scaling a network
(the more common one is using Router Reflectors)

Where, then, did the unordered sets (***_SET**) come from?

**First and foremost, this is the result of route aggregation**

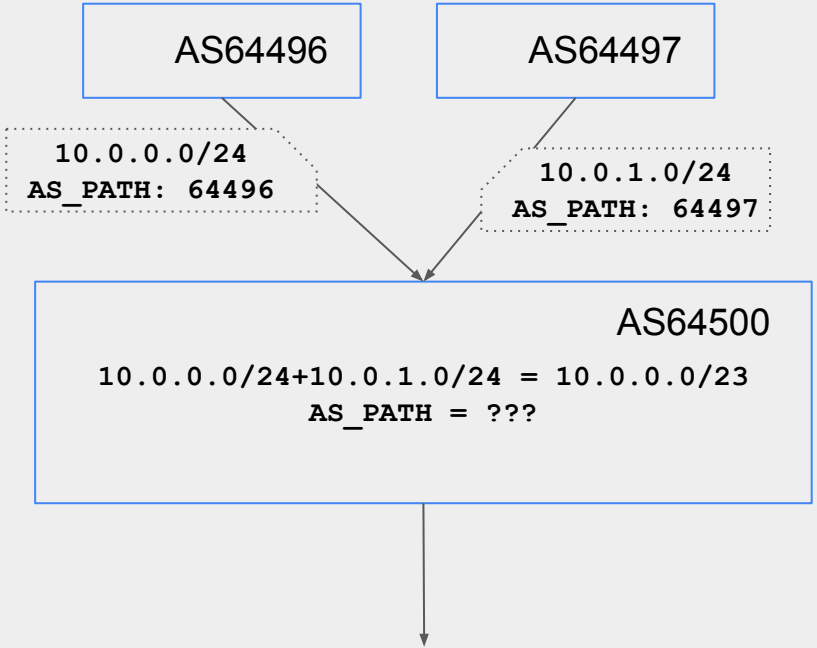
# Route aggregation in BGP

## Problem statement

It emerged in the 1990s, when the Internet's rapid growth was challenging router resources (memory and CPU)

Imagine the AS64500’s administrators want to reduce the number of routes that this AS announces

To do it, they decide to summarise the announcements from clients



# Route aggregation in BGP

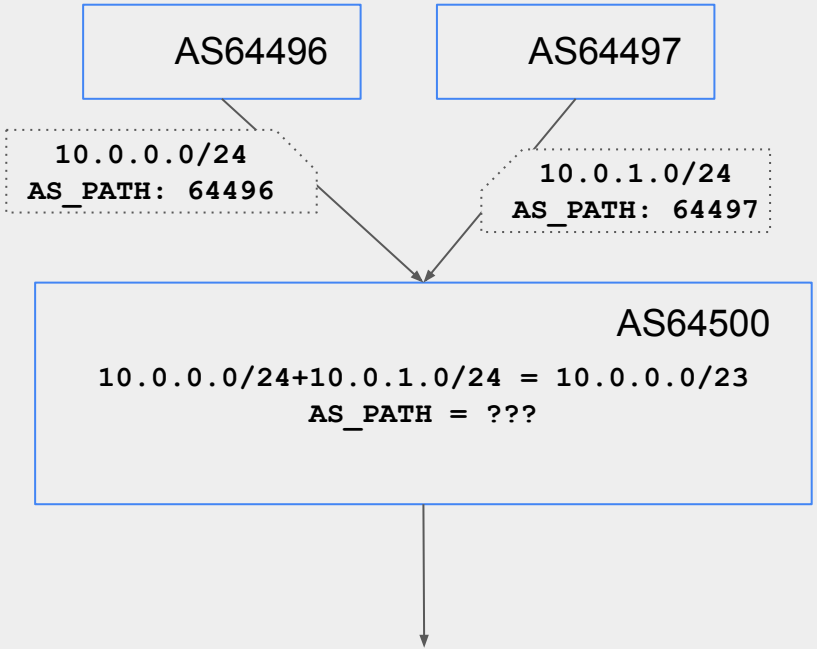
## Problem statement

The AS64500's administrators want to reduce the number of routes that this AS announces

To do it, they decide to summarise the announcements from clients

There are two approaches to do it:

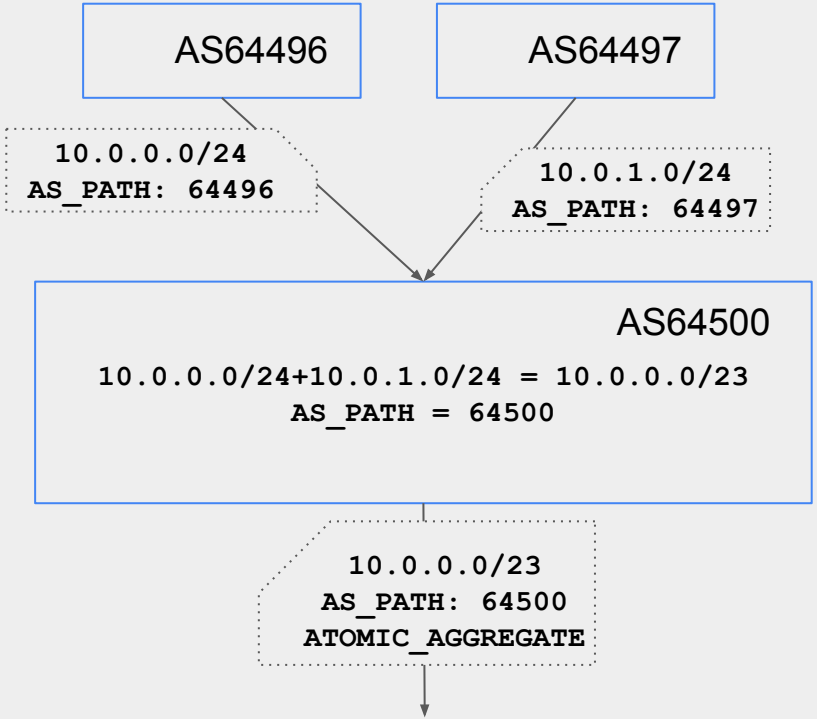
- Naive
- Data-preserving



# Route aggregation in BGP

## Approaches: naive

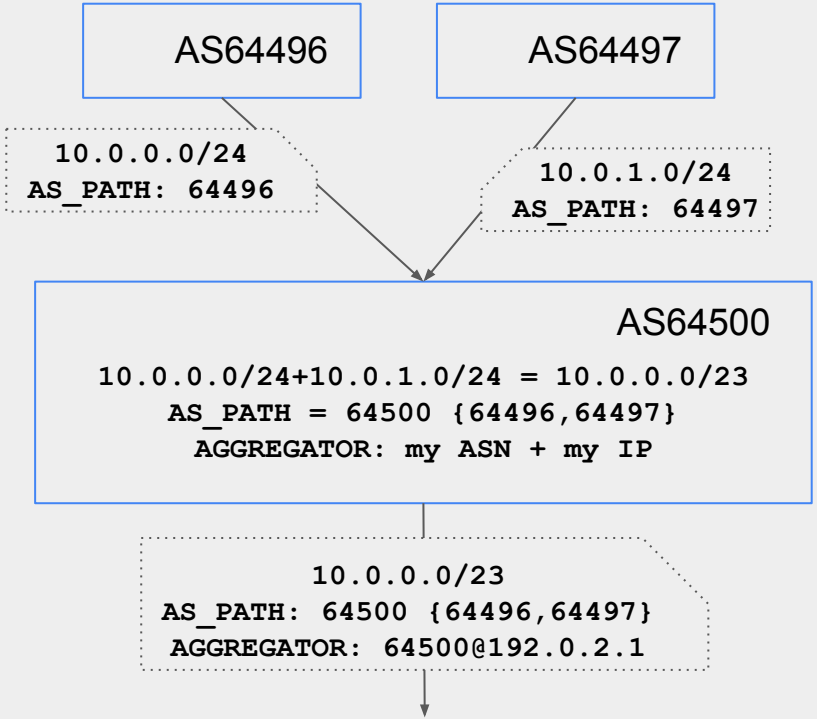
- Announce the aggregate itself, marking it with the ATOMIC_AGGREGATE attribute
  - "There was something else inside, but I won't say what"
- Simple, but it loses loop protection
  - If the aggregate returns to AS64496, the latter will accept the route announced by it



# Route aggregation in BGP

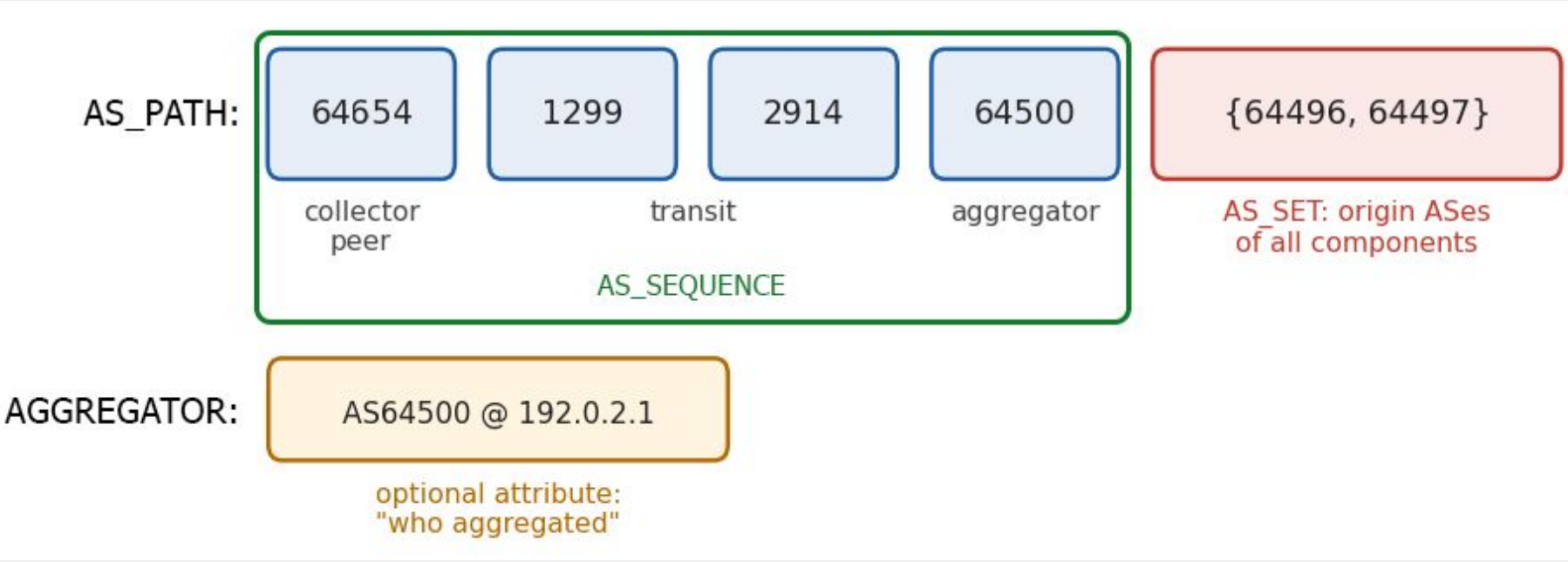
## Approaches: data-preserving

- Store the information as a set: **{64496,64497}**
- The order is lost, but loop detection still works: each participant will still find their number in the path
  - If the aggregate returns to AS64496, the latter will reject the route announced by it
  - An aggregating AS may identify itself using the optional AGGREGATOR attribute



# Route aggregation in BGP: data-preserving

## As it sees a RIPE RIS Collector



# History of AS_SET

## Withdrawal from standards

- **2011**: The IETF *recommended* against using *_SET
  - RFC 6472, using the "SHOULD NOT" directive

- **May 2025**: It *prohibited* them
  - RFC 9774 mandates that MUST NOT ("must not") routes with *_SET be advertised, and that received ones be treated as route withdrawals
    - **treat-as-withdraw**: the router behaves as if the prefix has been withdrawn

- The only quantitative snapshot we are aware of for these 14 years is the study by Hannachi and Sriram based on RouteViews data from **2019**
  - RFC 9774 itself cites this study: **477 routes with AS_SET** across the entire observable Internet, of which **383 are singleton sets**

# History of AS_SET

## Reality

Small and medium ASes did not need aggregations

For large ASes these techniques were a huge operational burden

**AS_SET turned out to be a construct that:**

- **almost no one uses**
- **almost no one filters**
- **almost no one understands properly**

It is precisely these kinds of constructs that, over time, turn into **a dusty attic where ghosts come to haunt**

# The Rationale for This Study

## Let’s watch a biopic

A single snapshot is a photograph

**We wanted a film**

What did the entire arc of AS_SET’s life look like,
from the heyday of the collective to its formal ban?

# Methodology

# Methodology

## Data sources

- Main source of data: RIPE RIS
  - 100 quarterly dates from 01/10/2001, to 01/07/2026
  - 48 hours windows
  - All available to the date RIS collectors

- CAIDA AS Rank

- RPKI historical data from RIPE NCC

# Step 1

## Extraction

Get all records and reduce them to

**<unixtime> <rrc> <peer_asn>@<peer_ip> <prefix> <aggr> <reduced AS PATH>**

**<unixtime>**: Unix timestamp of the first time this collector saw this exact tuple (`peer, aggregator, reduced path`) within the collection window

**<rrc>**: the RIPE RIS collector that recorded the route

**<peer_asn>@<peer_ip>**: the collector's peer via which the route was obtained

**<prefix>**: the prefix announced

**<aggr>**: value of the AGGREGATOR attribute (ASN and IP)

**<reduced AS PATH>**: An AS PATH in which consecutive identical ASNs (resulting from the prepending technique) have been replaced with a single one

# Step 2

## Canonisation

Output:

`<prefix> <n_paths> <aggr> <n_rrc> <pattern>`

`<n_paths>`: the number of different full paths that were resolved into this pattern

`<n_rrc>`: the number of different collectors that saw it

`<pattern>`: a pattern formed from a reduced AS PATH according to the following rules:

1. **Any contiguous sequence** of "non-standard" segments ({}, (), []) **is preserved verbatim**, along with **one ASN neighbor on each side**
2. The **last two path elements** (the origin side) are always **preserved**
3. Everything else is collapsed into *****

Deduplication key: tuple (`prefix, pattern, aggr`)

# Step 3

## Classification

Each line containing a non-standard segment receives some tags:

- The set is at the end of the path (is the origin) / in the middle
- Set size (1 indicates a **singleton**)
- The set contains private 16-bit and 32-bit ASNs, reserved ASNs, and unallocated ASNs
- The set includes Tier-1 operators
- Aggregator country
- Is the aggregator located directly to the left of its set along the path (aggregator matches left, **aml**)

Now any further analysis is trivial and computationally cheap

# Special case

## AML=0

Usually we have a number of an aggregating AS next to the aggregate:

```
AS_PATH = … X {A,B}, AGGREGATOR = X
```

If this is not the case, we have encountered an AML=0. Here are four theoretical scenarios how it can happen:

1. **Proxy aggregation (RFC 4271 behaviour)**: Autonomous System X aggregates routes received via a neighbour Y, which stays in the route as a common part:

```
AS_PATH = … X Y {A,B}, AGGREGATOR = X
```

2. **Full proxy aggregation**: on behalf of someone else. X configures the aggregation and, in addition, explicitly set a different ASN in the AGGREGATOR attribute
3. There is a private ASN between an aggregator and a set, which **by mistake** was not removed when the route was exported
4. Reaggregation without rewriting the attribute: Y receives the aggregate `X {A,B}` from X, with `AGGREGATOR = X`, aggregates it with something else into an even larger prefix, but **mistakenly** leaves AGGREGATOR untouched

# Who was aggregated?

## Analysis of the AS_SET content

1. We take the set, remove all private/service ASNs and the aggregator itself from it
2. For the remaining ones, look at the customer degree from the AS Rank database, and take the maximum: **maxdeg**
   a. The maximum, because a single large member means that the set is no longer just a “local party"
3. **Adjacent ratio**: the proportion of cases where maxdeg < 50

All the small members are most likely the aggregator’s directly connected downstreams

But if Lumen or NTT are present there, this is no longer an aggregation of its own customers

# Results

# Scale of the study

## Total across all 100 dates

Unique states (prefix, pattern, aggr): **67,124,177**

AS_SET segments: **29,421 (0.044%)**
AS_CONFED_SEQUENCE: **0 (0%)**
AS_CONFED_SET: **0 (0%)**

We could immediately wrap up this topic by saying,
"*no AS_CONFED*, and AS_SET is statistical noise*,"
but the rest of the story is all about
**what's hidden in that noise**

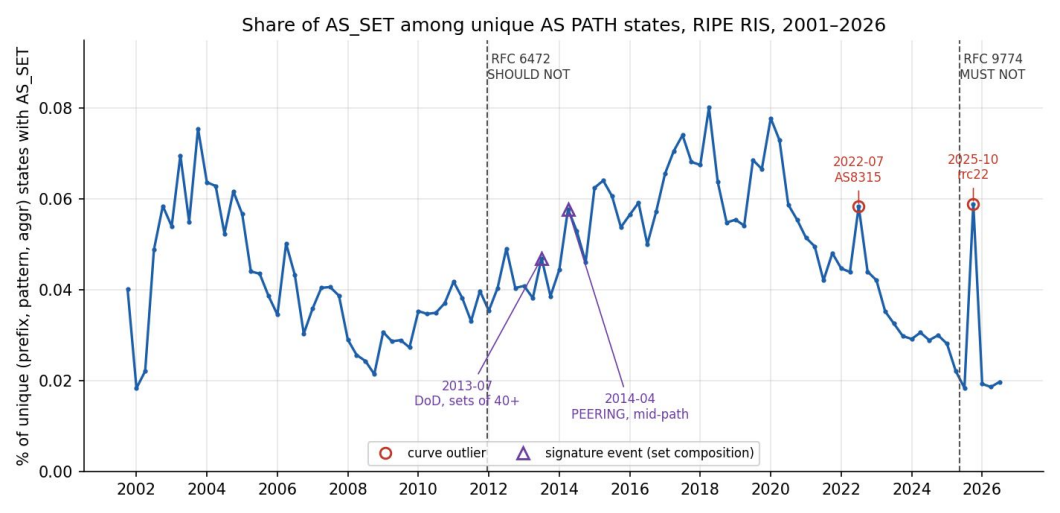
# AS_SETs in AS_PATHs

## Two waves

**The First Wave: 2002–2009**. Time of the official recommendation "**aggregate whatever you can**" (which few people followed anyway, peaking at **0.076%** in late 2003)

- 2008: "256k problem"

**Second wave, 2014–2020.** Era after RFC 6472, which "**SHOULD NOT**" recommendation was silently ignored. Absolute peak of **0.080%** in April 2018, second peak in early 2020.




Decline from 2020 to 2026 began five years before RFC 9774; in other words, the ban formalized a process that was already underway, rather than initiating it.

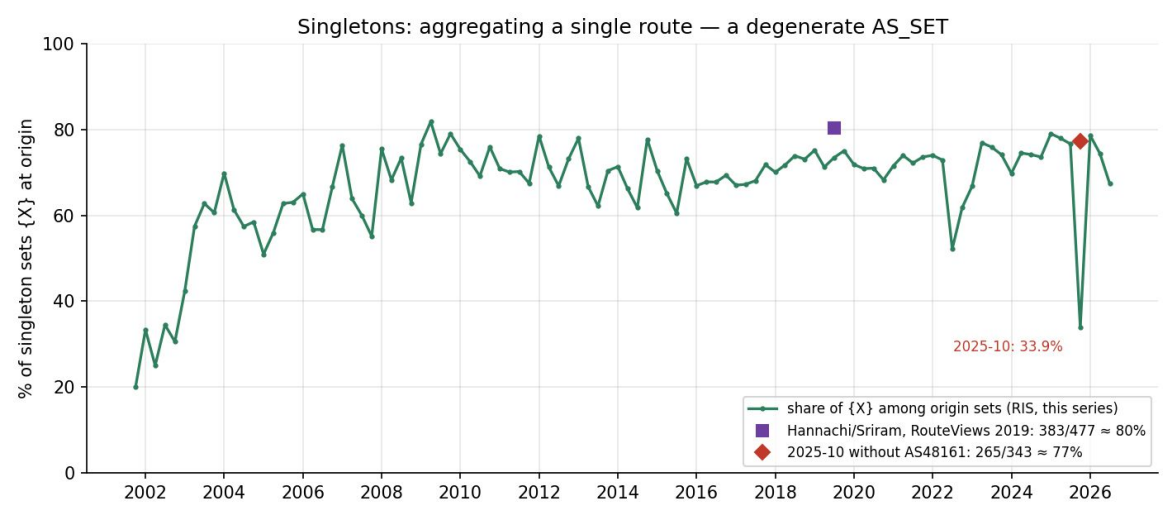
# Singletons in AS_PATHs

## Set of one ASN

Introduced to merge different paths, the set contains exactly **one element**

A singleton does nothing useful

It might be that just one route fell under the aggregation, and the hardware wrapped a single ASN in curly braces



# Singletons share

- From 2001 to 2003, singletons accounted for **20–40%**: during the first wave, AS_SET was still doing its job (when used)

- But starting in 2009, the percentage leveled off at **70–80%** and has remained there for **17 years**
  - Our measurement for July 2019 (476/648 ≈ 73%) is consistent with the independent measurement by Hannachi/Sriram on a different set (383/477 ≈ 80%)
    - That very measurement, which is cited by RFC 9774 for the recommendation to stop using AS_SET

- The remaining quarter (of a set with 2 or more elements) is where everything else occurs

# AGGREGATOR attribute

## Who is responsible for the aggregation?

- About **7%** of routes carry the AGGREGATOR attribute (2026-07 snapshot)
  - It doesn't require an AS_SET on its own and most often appears without one
  - It's simply a tag indicating that “this prefix was somehow aggregated along the way”
- The good news: throughout the entire series, there wasn’t a single unregistered ASN in the AGGREGATOR field
- The bad news: as of today, about **2 million records (roughly one-third)** carry a **private ASN** in the AGGREGATOR field
- Record holder in our sample: AS65367, aggregating 5,326 prefixes from a single large operator (Optus)
  - It doesn't break routing; it just shows to the whole world the intimate details of the operator's inner functioning

# Who was aggregated?

## Analysis of the AS_SET content

- A "normal" country is **80–100% adjacent AS_SET**
  - "Adjacent AS_SET" means that it includes only nearby customers
  - Aggregation is local "backyard" operations, bundling their own customers

- And two glaring outliers: the **Netherlands** (41%) and **Romania** (35%)
  - Most of the aggregation appears to be remote!
  - A systematic pull into the AS_SET of distant large networks?
    - We'll see later!

**Local/remote aggregation, Top 10**

| CC | Entries | Adjacent ratio |
|---|---|---|
| US | 7973 | 89% |
| IN | 1430 | 94% |
| RU | 1082 | 91% |
| CA | 1073 | 91% |
| **NL** | **879** | **41%** |
| GB | 802 | 80% |
| EG | 738 | 100% |
| **RO** | **727** | **35%** |
| UA | 720 | 95% |
| BB | 651 | 100% |

# Yet another evolution

## 32-bits ASNs and AS_TRANS

- **AS_TRANS is AS23456**, a relic from the time when the industry was transitioning to 32-bit ASNs from 16-bits ones
  - This ASN was used to replace 32-bit ASNs in UPDATE attributes
  - Specifically: in **AS_PATH** and **AGGREGATOR**
    - Shadow attributes: AS4_PATH and AS4_AGGREGATOR
  - Now it is **ASN-bogon**

# Yet another evolution

## AS_TRANS: two attributes, two roles

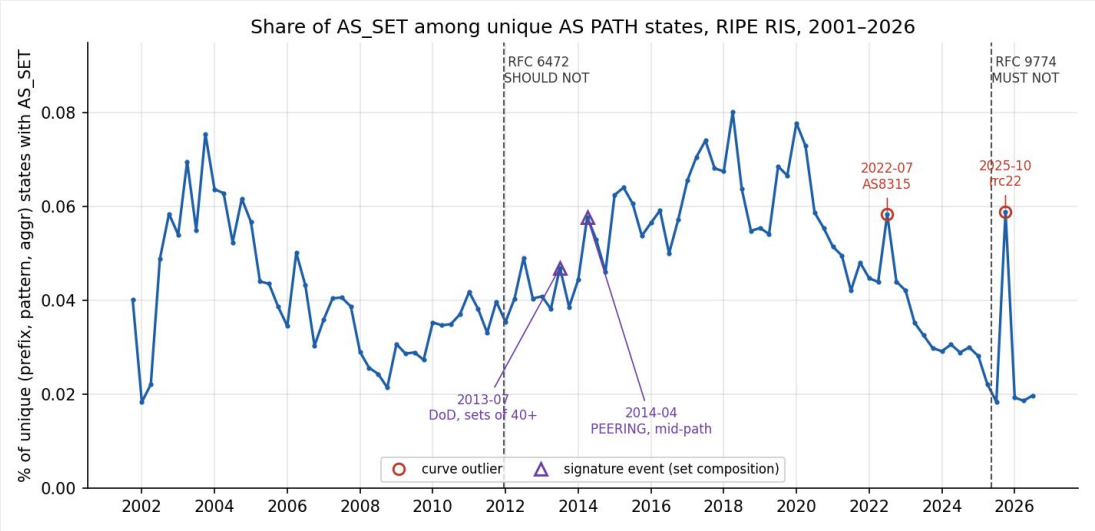
- **First role: AS_TRANS is in the path itself (1432 entries)**
  - Moreover, in 1,413 cases, this is the **origin**
  - Checked on all such 15,861 prefix-dates: **96.6% are shadowed** (32-bit origins whose AS4_PATH was lost on one path)
  - 520 (3.3%) have no other origin anywhere; **229** of them are seen by ≥5 collectors — the best **candidates for genuine**
  - **291 AS_SETs are a singleton {AS_TRANS}** (but a singletons is redundant by itself)

- **Second role:** the **AGGREGATOR field contains it**, but there is **no shadow AS4_AGGREGATOR (351 entries)**
  - Source: a certain transit router on the link through AS2603 (NORDUnet) for a 16-bit neighbour generated an AS4_PATH but not an AS4_AGGREGATOR
  - Half of the transition mechanism was implemented; the other half was forgotten
  - It is impossible to restore the aggregator: the information has been destroyed

# Back to the graph of AS_SET usage

## Four different stories

- **The curve itself revealed two** events (peaks in **July 2022** and **October 2025**)
- Two were found based on the composition of AS_SETs:
  - **April 2014**, AS_SETs in the middle of an AS_PATH
  - **July 2013**, AS_SETs with four dozen members



# April 2014: PEERING testbed

## AS_SET in the middle of AS_PATH

- Most of the output lines do not have an AGGREGATOR attribute at all
- A set not at the origin - that in itself is strange
  - A “wandering” set in the middle means that someone is manually constructing the path
- This is the first and the largest instance in 25 years of a spike in sets in the middle of the path
  - Did not observe before
  - Has persisted as a background of 1–16 entries per quarter ever since
    - PEERING + Voxility Traffic Engineering mechanisms + noise

# April 2014: PEERING testbed

## Dive deeper: who?

- The prefix **184.164.240.0/24** announced is from the 184.164.224.0/19 block belonging to **PEERING - an academic testbed, AS47065**

- The AS_PATH itself is a textbook "sandwich": … **47065 {174,11537} 47065**
  - The origin is repeated on both sides of the set containing Cogent and Internet2
  - This is AS_PATH poisoning: researchers are deliberately cutting off networks from the prefix's distribution zone

# April 2014: PEERING testbed

## Dive deeper: why?

- AS_SET here serves as a convenient container for these “poisons”

**These are research UPDATEs: legitimate,
but unrelated to the “natural” use of AS_SET**

Since it was “caught” during traffic spikes, an additional conclusion follows: by 2014, **AS_SET had already become more interesting to researchers** as a tool for manipulation **than to operators** as a tool for aggregation

# July 2022: Chronicle

## What is so special about this date?

- The singleton share drops to 52%
- priv32 jumps from a few units to 266
- AML=0 spike: 285 of 716 (40%)

And some private ASNs in AS_SETs has … structure?

`4207220701 → 4207-22-07-01`

Four of the eight encode a date; one of them — the day it first appeared in the table

These ASNs are visible through the AS8315 aggregator, a Dutch hosting provider

| Ent. | ASN | Notes |
|---|---|---|
| 264 | 1299 | ← Arelion |
| 263 | 4207220701 | |
| 263 | 4207220601 | |
| 259 | 4207100000 | |
| 258 | 4207710101 | |
| 258 | 19994 | ← Rackspace |
| 258 | 12200 | ← Rackspace |
| 255 | 4207720101 | |
| 235 | 4207211001 | |
| 217 | 4207210601 | |

# July 2022: Chronicle

## What happened

- As of this date, AS8315 has two aggregate prefixes
  - ~130 different set compositions for each
- The public core (1299, 19994, 12200) appears in nearly every record
  - The eight private ASNs vary: out of 256 possible subsets, ~130 observed
- Aggregation was designed to hide internal instability, but…
- Here, aggregation with as-set and flapping private ASNs does exactly the opposite
  - Every internal fluctuation changes the aggregate's attribute
  - Internal instability is exported to the global table in the form of a change to AS_SET
- Three-level hierarchy — 8315 → 28836 → private sub-ASs; an AML=0 spike in 285 entries is caused by this internal level, which has not been removed from the path
- Additionally, AS8315 combined some external learned routes with its own internal ones

# July 2022: Chronicle

## What happened

The hosting provider used private ASNs generated based on the date
(when the site went live?)

Because there is no common mechanism in AS_SET to filter out private ASNs,
it exposed the result to the outside world

For 9 months, AS8315 broadcast not only
its own numbering but also the pulse of its sites to the world

# October 2025: Bucharest

## Local leak

- The entire event is visible through exactly one peer-to-peer session of a single collector: **rrc22 from AS48571**
  - Not “primarily”, but every single path

- **AS_SET: the major Romanian operators** (AS12302 Vodafone Romania, AS8708 DIGI/RCS&RDS, AS8953 Orange Romania, AS6663, AS5606) intermixed with an **Iranian cluster** (AS12880 DCI, AS42337 Respina, AS43754).
  - The setup remained active for at least two weeks as a regular one

- A regional player was using **proxy aggregation**
  - By pulling half of Romania, a portion of Iran, and transit networks into its AS_SETs
  - Other RIS collectors didn't see this, because the setup didn't extend beyond a single peering point

## Conclusions

The collector saw something no one else in the world had seen, because it happened to be in the right place (just one hop away from the source)

**Your local vantage point is both a privilege and a bias**

It’s also a perfect illustration of why it’s absolutely essential to **filter what you send and receive in BGP**

# July 2013: Disclosure (DoD)

## An anomaly from the table

- As of July 2013, there were 872 public AS_SET members, compared to 412 and 287 for neighboring dates
  - Fifteen entries with AS_SETs containing 38–44 members
  - Among the largest AS_SETs from the first twenty years of the series,
  - All fifteen are tagged by the AS721 aggregator
  - Members (AS83, AS523, AS575, AS668, AS1527 …): ASNs from DNIC blocks

- Visibility: up to 13 collectors
  - In April 2013 and October 2013, AS721 had no AS_SETs at all

- This is not a leak: the numbers are public, the AGGREGATOR attribute is used
  - So no filter would have (or should have) stopped this

# July 2013: Disclosure (DoD)

## Conclusions

**This seems like a voluntary disclosure of the internal structure**

A "**confederation from improvised means**" on the largest scale,
but with public rather than private sub-ASs

# Four different stories

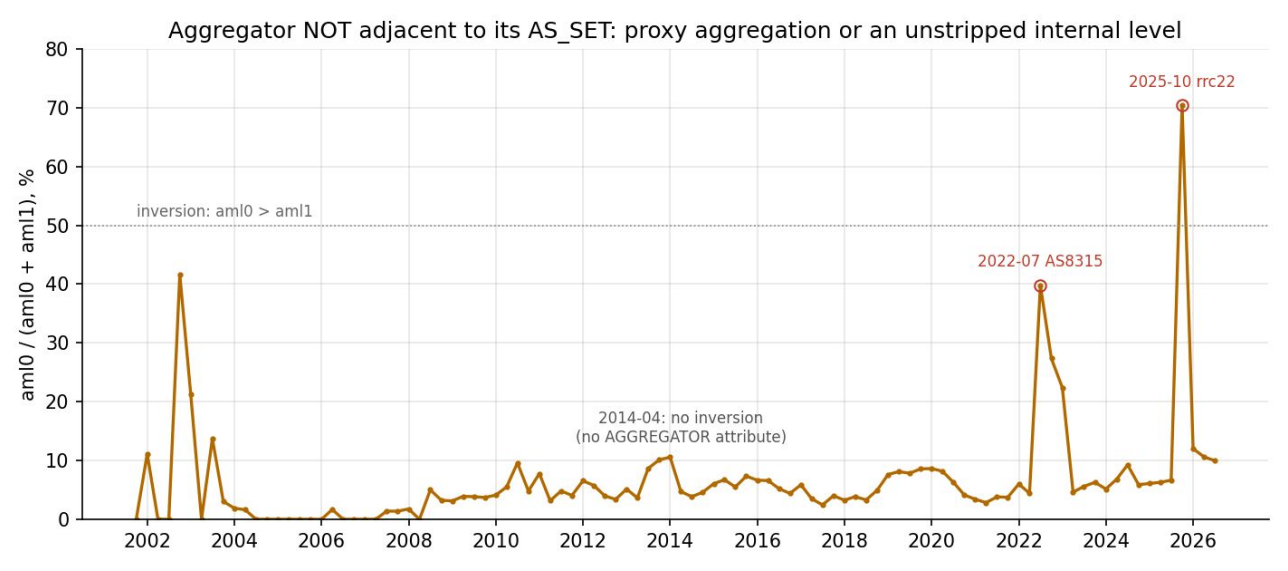
## Main observation

Four events, four distinct mechanisms
and zero instances where AS_SET exactly does
the job it was designed to do

# AML again

## More outliers

- 2002-2004: The small numbers effect
- PEERING case is under radars
- 2022-07 and 2025-10 are worth to analyze



# AML again

## 2022-07: proxy aggregation for the own sibling

We discussed the AS8315 case earlier

**AS8315 aggregated routes learned from AS28836 (same organisation), which stayed in the sequence**

The private ASNs are a separate leak:
inside the set, not in front of it

# AML again

## 2025-10: Aggregation on behalf of a client (proxy aggregation)

720 entries from AS48161: to the left of its AS_SET are only third-party transits

- AS9050 (373 entries)
- AS49666 (74 entries)
- AS31287 (32 entries)
- followed by nine more Romanian and regional networks, each with several entries

# Private ASNs in AS_SETs

## What if there are only private ASNs?

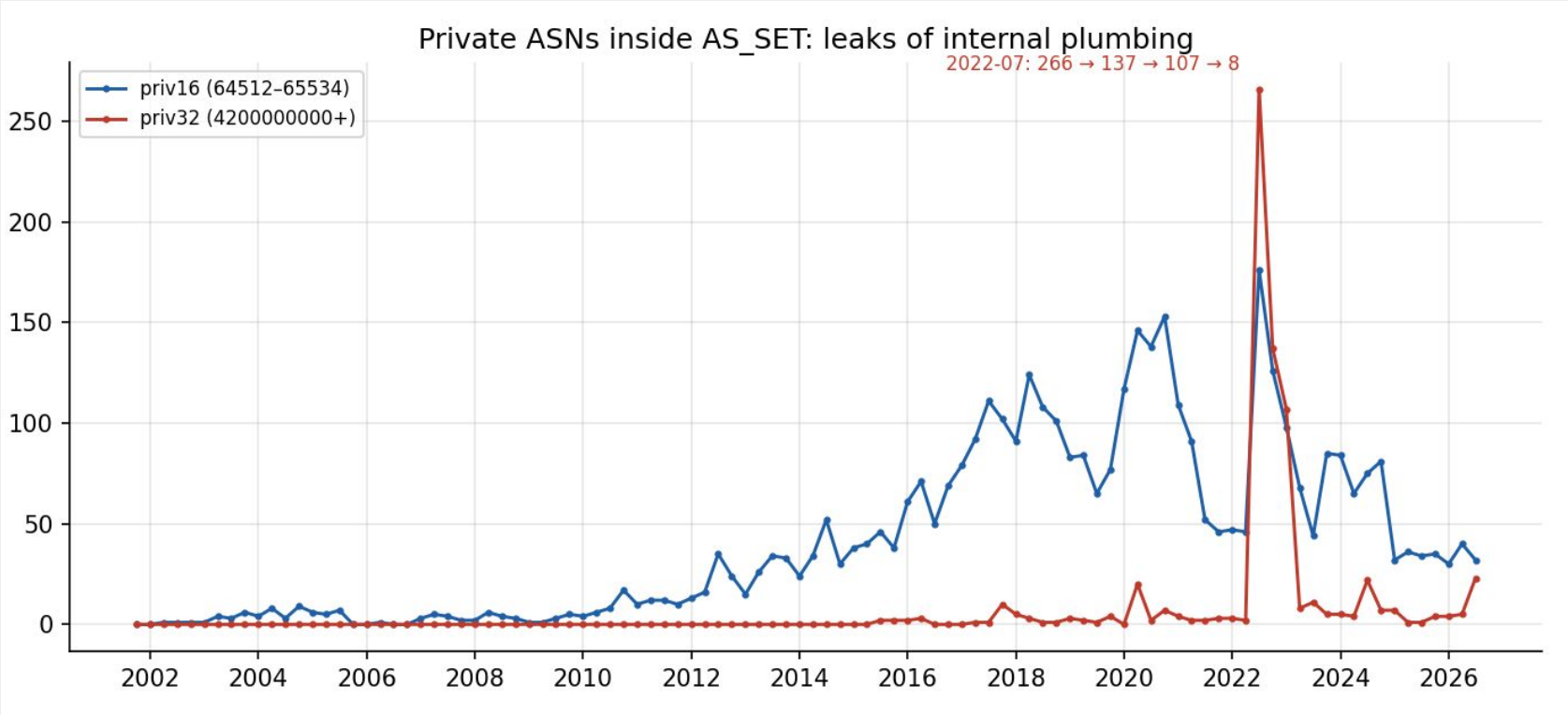
It looks like a simple substitution of AS_CONFED_SET

- Private ASNs for internal domains, plus standard eBGP between them
- Vendor manuals have been recommending this exact approach for twenty years
- Only one hygiene step has been omitted

The hypothesis yields three testable predictions:

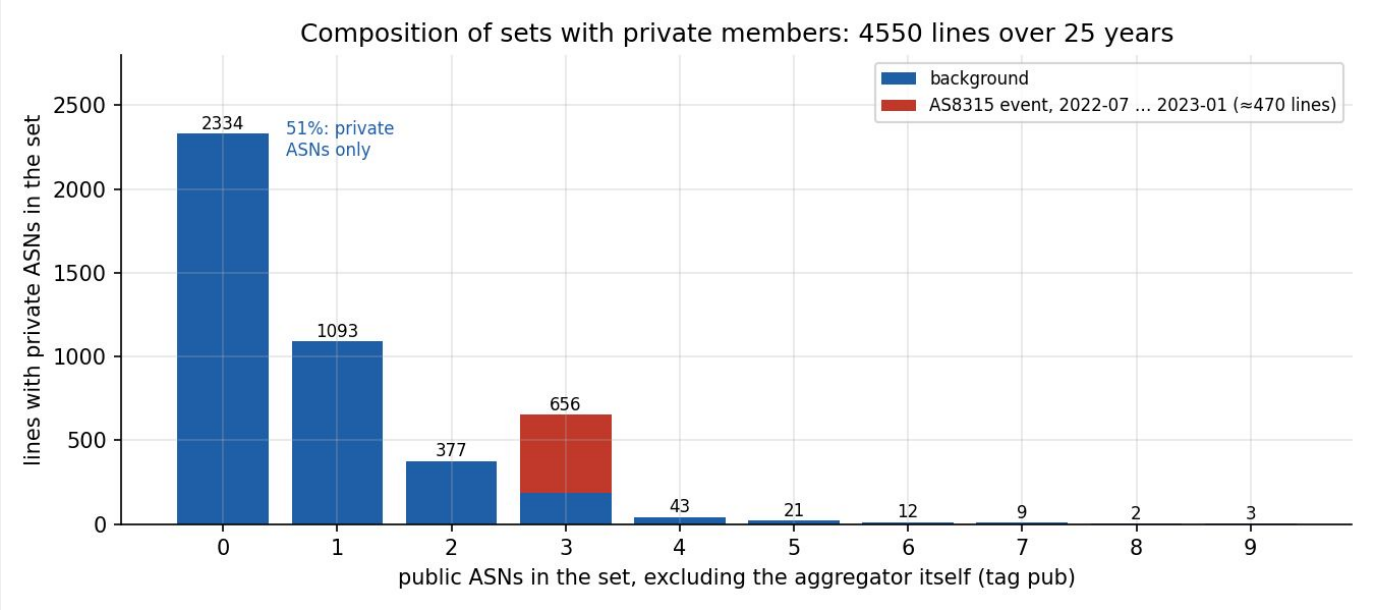
1. in such sets there are almost no public ASNs (but the aggregator itself),
2. this practice is long-standing (it is an architectural feature, not a random incident),
3. private numbers are not random (i.e. there is a numbering plan).

# How often do we see them?



# Private ASNs in AS_SETs

## Proof of prediction number 1



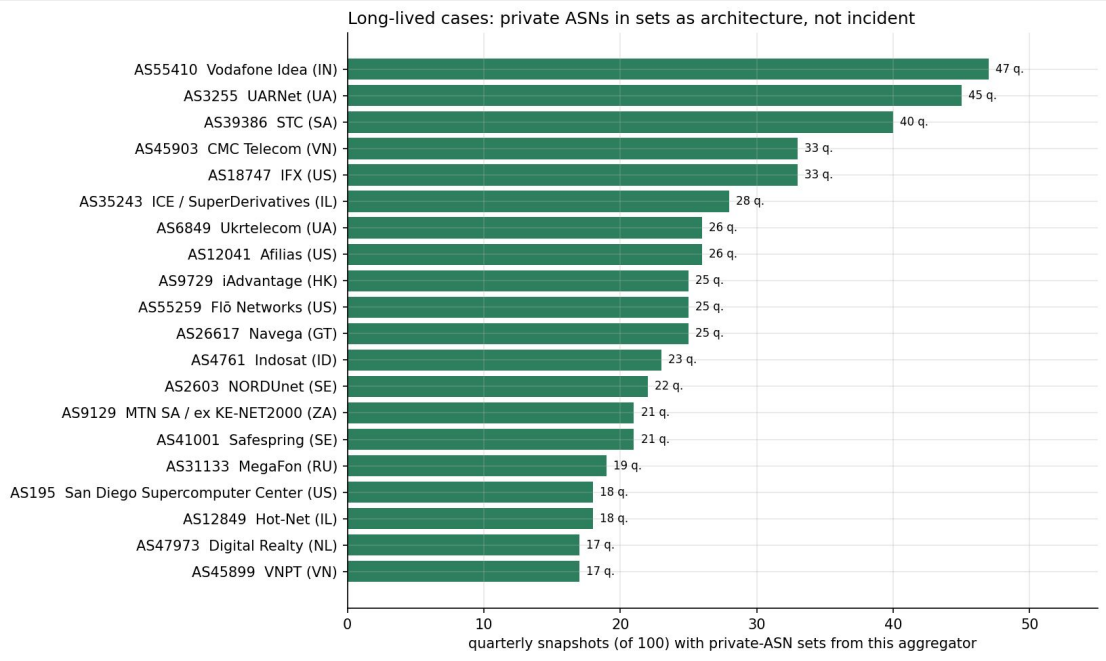
# Private ASNs in AS_SETs

## Proof of prediction number 2

These are clearly **architectural designs**

The list consists of regional or mostly national-scale operators

That is exactly the weight class for which the confederations were invented



# Private ASNs in AS_SETs: particular cases

## AS23966 (Pakistan): prediction 3

- Observed for eight quarters from 2019 to 2021
- Its AS_SET includes: AS45779 (interbank communications network), and dense **clustered blocks** of private ASNs: 64592, 64601, 64604, 64616, 64547, 64863, 64900, 64905
- The architecture is two-tiered: a banking network on private sites plus an upstream provider that aggregates it with as-set
  - And has been broadcasting its numbering plan to the world for two years

# Private ASNs in AS_SETs: particular cases

## AS9441 (Bangladesh)

- The same two-tier structure: public clients (AS58805, AS135037, AS134180…) and, next to them, private clients 64562–64596
- A single prefix exists in two states simultaneously
  - Say, for 103.4.144.0/22, 17 collectors see `AS_PATH = {58805}` and 14 see `AS_PATH = {58805,64575,64584,64595}`

# “Missing Digit” Error Class

## AS202896: “4 | ASN | PoP” plan: prediction 3

Snapshot from July 1, 2024: AS202896 had AS_SETs packed with 32-bit private ASNs with a transparent structure: 4202896005, 4202896007 … 4202896047

It reads as

**`4 202896 NNN = “4, own public ASN, PoP number”`**

An elegant, self-documenting numbering scheme. What could have possibly gone wrong?

Two sites (8 and 9) have one leading zero instead of two

With 9 digits instead of 10, the number falls outside the private range and into unassigned space: **4 202896 08** and **4 202896 09**

It is worth noting that these ASNs also appear in AS PATH as regular origins, not as part of an AS_SET

# AS202896: homemade confederation

## Operator hierarchy

The three-level hierarchy reconstructed from the sets is as follows:

```
202896 {65288, 64999, 65180, 65182, 4202896005, …} ← top level
65288 {64999, 65180, 65182, 4202896005, …} ← without 65288
64999 { 65180,65182, 4202896005, …} ← without 65288 and 64999
```

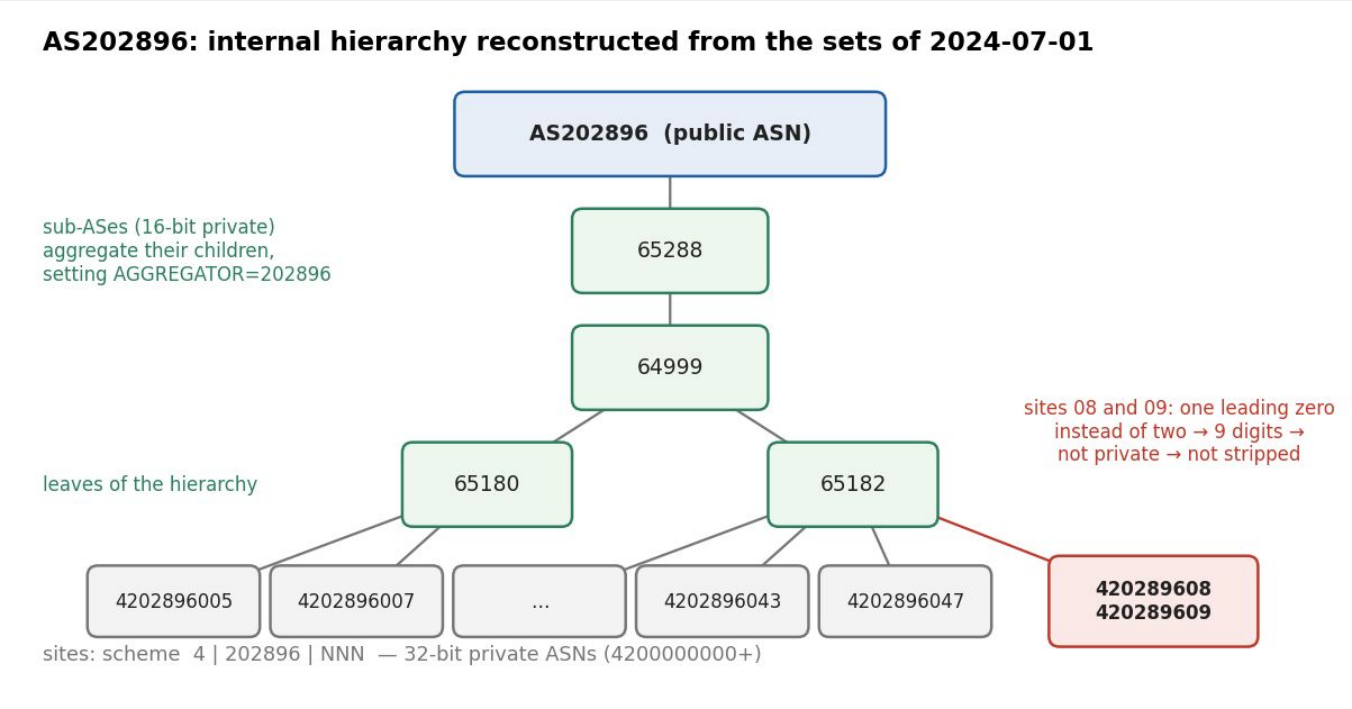
Each sub-AS aggregates its children and sets AGGREGATOR=202896:

- **a confederation with a member-AS hierarchy**
- Only without mandatory scrubbing at the border
- Which is what allows us to understand this structure

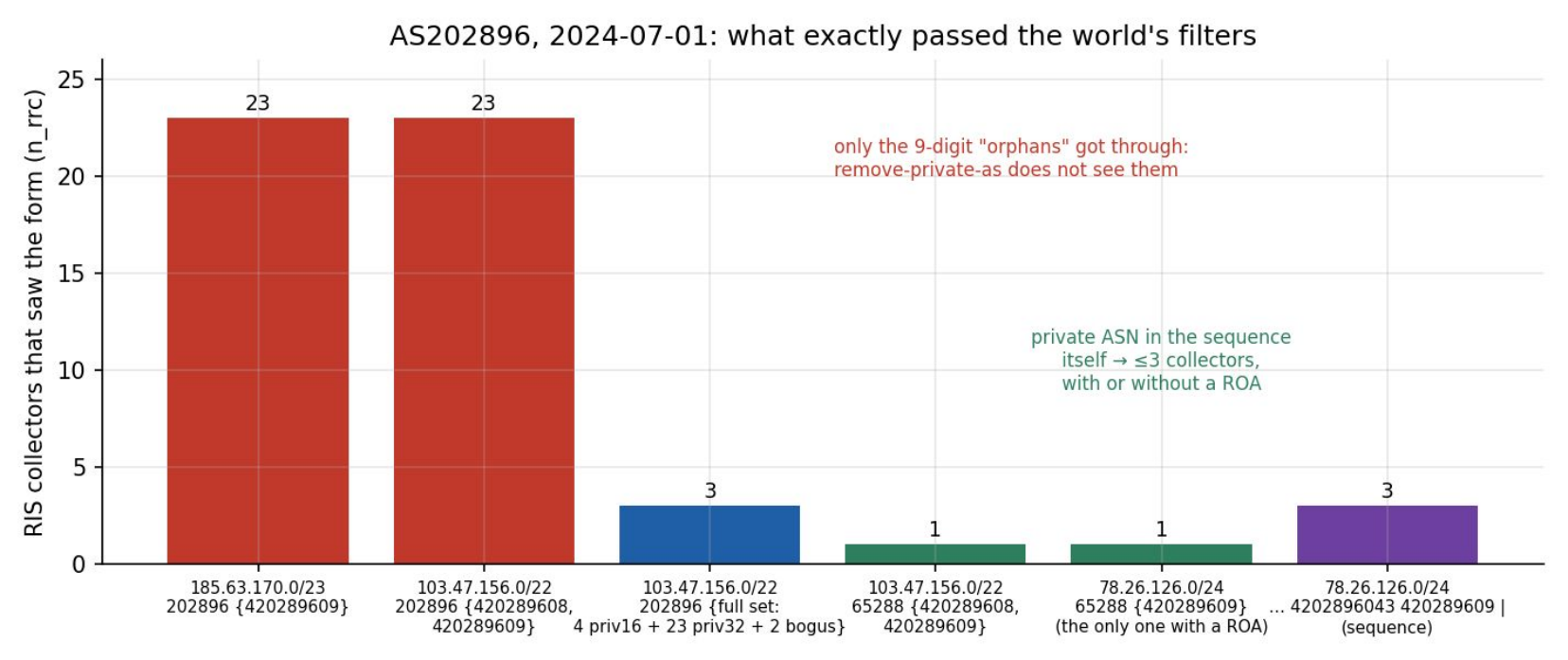
The situation lasted no more than a quarter

# Internal structure of AS202896

## As restored from AS_SETs (2024-07-01)



# How it is seen by RIS collectors



# DoD: second (unintentional) exposure (not in AS_SETs)

## July 2025 through August 25, 2026: who?

Another leading zero was missed

AS_PATH for 143.68.204.0/23:

- **three public DoD ASNs** (AS721, AS27064, AS1540)
- followed by **eight unassigned ASNs: 421005091-094 and 421005151-154**

# DoD: second (unintentional) exposure (not in AS_SETs)

## July 2025 through August 25, 2026: what happened?

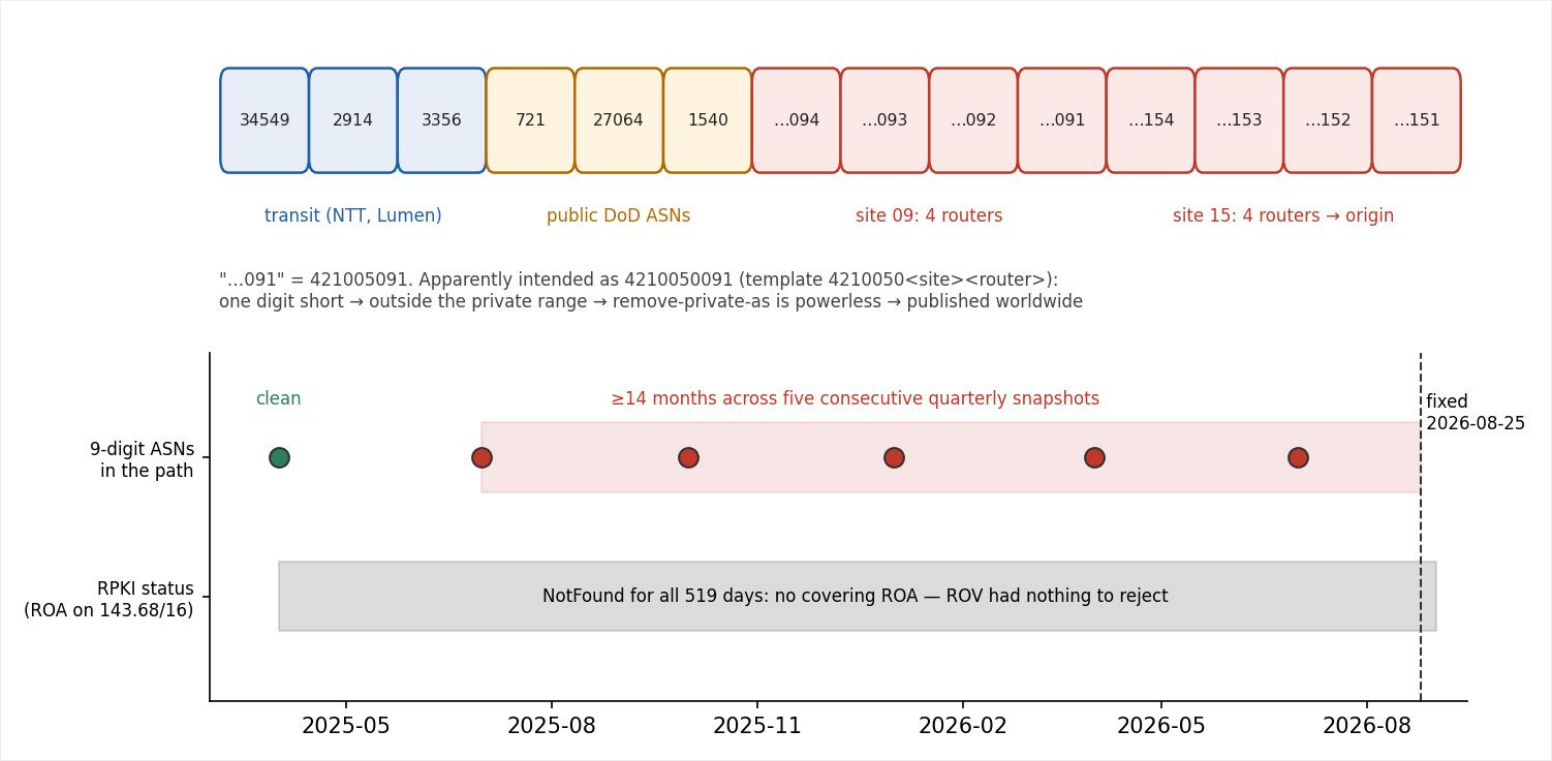
**The numbering plan** was leaked (template **4210050<site><router>**, two sites with four devices each)

The internal eBGP topology was leaked: **two sites connected in consecutive order**, with **routers inside linked in a "daisy chain"**

Since this is an AS_SEQUENCE, presumably the properly formatted private fields have been removed everywhere and are no longer visible anywhere; only the typos remain

- For the same reason, these erroneous ASNs ended up in the AS Rank (CAIDA)

# Internal DoD structure for that period of time



# Private ASNs in AS_SETs

## Homemade AS_CONFED_SET hypothesis

Three predictions were **confirmed**

However each prediction was **confirmed on a different set of cases** — no single case was checked against all three

So, while **this cannot be considered rigorous proof** of the hypothesis that AS_SET is often used as a simple substitution for AS_CONFED_SET, **it does lend significant weight to its validity**

# Conflict between AS_SET and BGP security models

## Both ROV, ASPA and BGPSEC break down

AS_SET at the beginning of AS_PATH (right-most segment)

- **The set becomes the origin AS**
- But there are **no sets in ROA**, and if an ROA record exists, the path becomes **INVALID**
- RFC 9774 explicitly mentions this as one of the reasons for prohibiting AS_SET

AS_SET anywhere in AS_PATH

- The **set becomes a neighbor** in the path, but there are **no sets in ASPA**
- A strict ASPA check is not possible
- A partial check may overlook violations of ASPA principles

AS_SET is not supported by BGPSEC at all

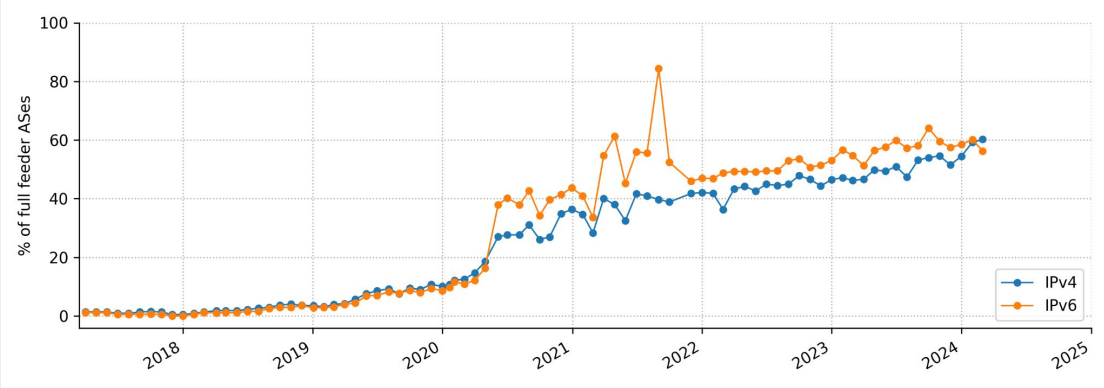
# RPKI and the second wave shape

## The decline after 2018

AS_SET does not interfere with ROA creation (signing phase)

But everything breaks down during the validation phase (RoV)

The growth of RoV began in 2019




https://ctestart.github.io/BGP-RPKI-ROV/

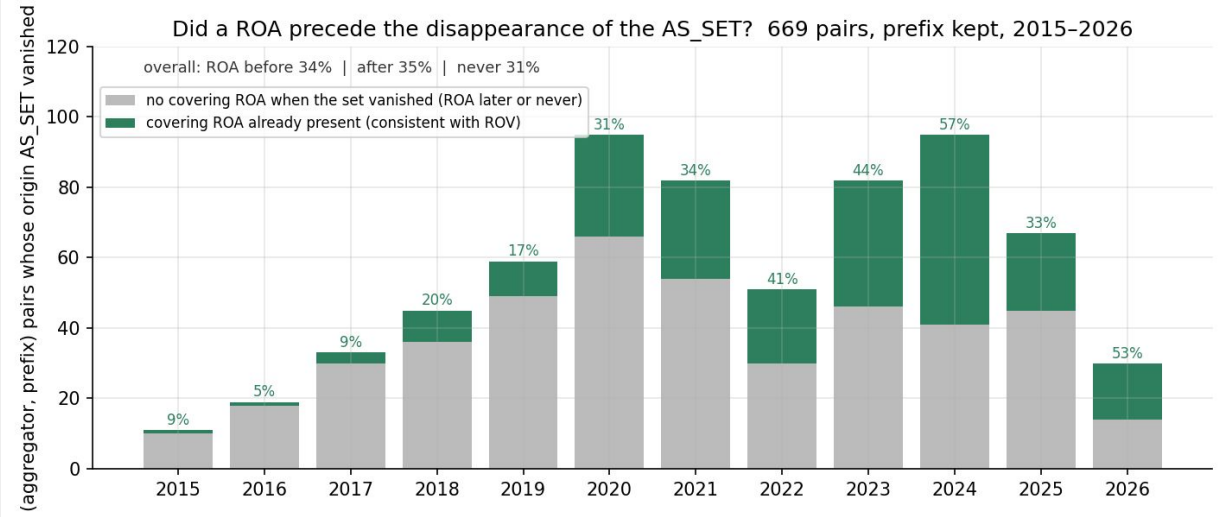
It is tempting to assume that the decrease in the AS_SET share is due to the growing adoption of RoV…

# Did ROV kill AS_SET?

## Not exactly

Of the 669 AS_SET disappearances analyzed (where the prefix was preserved), 439 (**65.6%**) **did not have a covering ROA at the time of disappearance**

- 233 received one later
- 206 have not received one until now




**Direct invalidation in ROV does not explain most of the observed decline**
**Its contribution is at most one-third (and more likely less)**

# Future study of AS_SET usage declining?

**Probably not**

## The real reasons are not completely clear yet

We've examined one hypothesis (RPKI) that has direct operational significance

- It's quite possible that additional hypotheses could be formulated
- But they would most likely be of only academic and historical interest, so I've stopped here
- If anyone finds anything else interesting that I've overlooked - that would be great

In any case, there is now an explicit RFC ban on the use of AS_SET, and the only question is when this mechanism will appear in router firmware

# Conclusions

# Conclusions

## At a glance

- AS_SET, although it has made its way into all the textbooks and vendor tutorials, has remained **a stillborn mechanism**
- Of what is available, **three-quarters are singletons**
- The remaining **quarter consists almost entirely of byproducts**: the internal structure of networks, experimental manipulations, proxy aggregation of third-party routes, and so on
- **There are (and have been last 20 years) very few cases where AS_SET actually performs the task for which it was designed**

# Conclusions

## Why so?

- When a mechanism is not widely used, neither operational experience nor the identification of implementation errors occurs
  - **Rare use leads not to benefits, but to oddities**
- The mechanism interacts with others, and the result can be confusing (private ASNs, AS_TRANS) or harmful (RPKI)
- Every backward-compatibility mechanism introduces a new class of errors, and it outlives the transition itself
- **A non-mandatory recommendation (SHOULD NOT, 2011) had no effect** (let's see what happens with MUST NOT)
- The decline in usage has not been fully explained, but this insight is hardly important

# Conclusions

## Some consequences of using AS_SET

- Leaks of the internal structure through the mechanism that is supposed to hide it
- Tier-1 entries end up in someone else's AS_SET, while internal flaps are exported to the global table
- AS_SET protects errors (such as formatting ones) from being detected

# Conclusions

## Main message

**The KISS principle applies fully to the formation of AS paths**

RIPE NCC
RIPE NETWORK COORDINATION CENTER

# THANK YOU!

# Questions & Comments

?

asemenyaka@ripe.net