RIPE NCC
RIPE NETWORK COORDINATION CENTER

Наша пісня гарна, нова

Або знову про BGP

Трохи згадок про минуле

Початок BGPv4

Кінець 1980s – початок 1990s

Інтернет був маленький,
а довіра -повною

Але вже було очевидно, що потрібен універсальний та глобальний механізм побудови маршрутів (про безпеку жодного слова не йшло)

Щоб вирішити цю задачу, Інтернет поділили на автономні системи, котрі спілкуються одна з одною через відкритий протокол BGP

Минуло багато часу...

А BGPv4 все ще з нами

Великій успіх через універсальну архітектуру

Але він весь час розвивався, та продовжує розвиватися

І цей розвиток не завжди має якусь логіку розвитку

Чому?

Минуло багато часу...

А BGPv4 все ще з нами

Великій успіх через універсальну архітектуру

Але він весь час розвивався, та продовжує розвиватися

І цей розвиток не завжди має якусь логіку розвитку

BGP виглядає як лоскутна ковдра через те що це гроші, котрі переклали на мову маршрутизаторів

Що таке BGP зараз?

Універсальний механізм

BGP - розподілений засіб горизонтального обміну інформацією, котра залишається стабільною протягом певного часу

Частіше за все це маршрутна інформація, але це не обов’язково

Безпека у BGP

Тут не все гаразд навіть зараз

Також, у оригінальному протоколі не було нічого про безпеку

Бо нащо, якщо всі всіх знають?

Тому протокол у тому погляді як він був створений, не відповідає поточним вимогам безпеки та операційним стандартам

Почнемо з безпеки та економіки бізнесу

Термінологія

Крадіжка префікса та витік маршрута

Крадіжка префікса (prefix hijack)
AS анонсує префікс, котрий не має прав анонсувати

Витік маршрута (route leak)
Розповсюдження префікса шляхом, котрим воно не має відбуватися

**Можливі інші сценарії, скажімо,
через модифікацію різних атрибутів BGP**

Як часто трапляються інциденти?

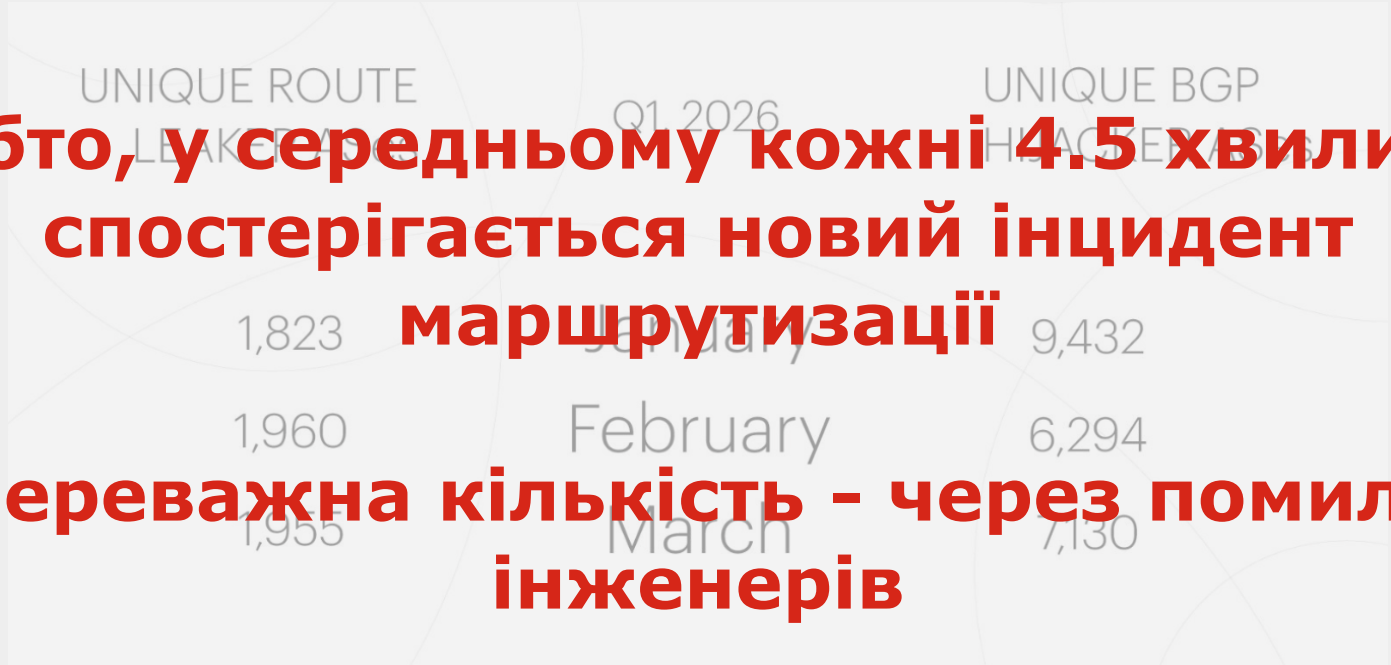
Статистика по типах

UNIQUE ROUTE LEAKER ASes	Q1, 2026	UNIQUE BGP HIJACKER ASes
1,823	January	9,432
1,960	February	6,294
1,955	March	7,130

Data from https://qrator.net/blog/details/Q1-2026-DDoS-bad-bots-and-BGP-incidents-statistics-and-overview/

Як часто трапляються інциденти?

Статистика по типах



Тобто, у середньому кожні 4.5 хвилини спостерігається новий інцидент маршрутизації

І переважна кількість - через помилки інженерів

Data from https://qrator.net/blog/details/Q1-2026-DDoS-bad-bots-and-BGP-incidents-statistics-and-overview/

А тепер про щось геть інше

Money talks

Так, BGP - це протокол маршрутизації, але не будь якої. BGP має бути протоколом маршрутизації, котра заробляє операторам гроші

30 років BGP вдосконалювався так, щоб допомогати їм купувати та продавати трафік

BGP = “гроші, котрі переклали на мову маршрутизаторів”

Неможливо говорити про BGP у Інтернеті, не пам’ятаючи про бізнес

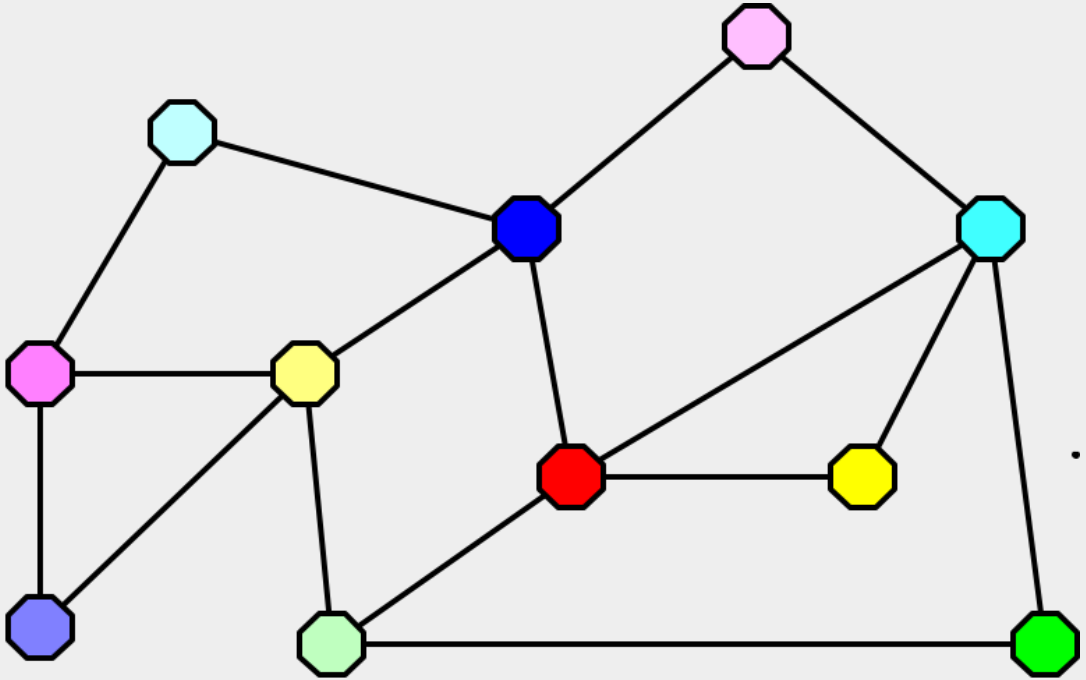
Як все починалося

Достатньо хаотичні з’єдання

Випадковий обмін трафіком

Складні взаємодії

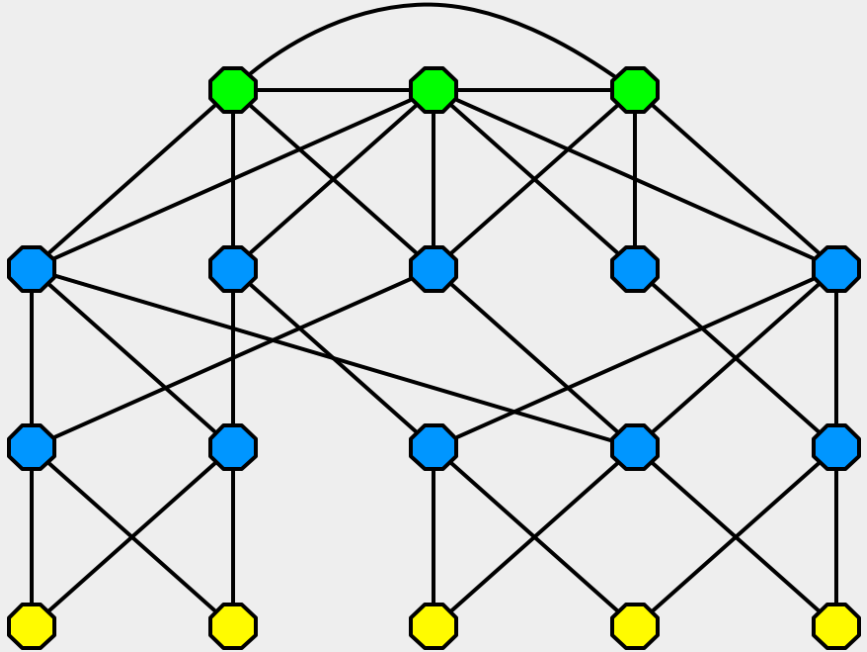
Часто некомерційні



А зараз що?

Комерційна ієрархія

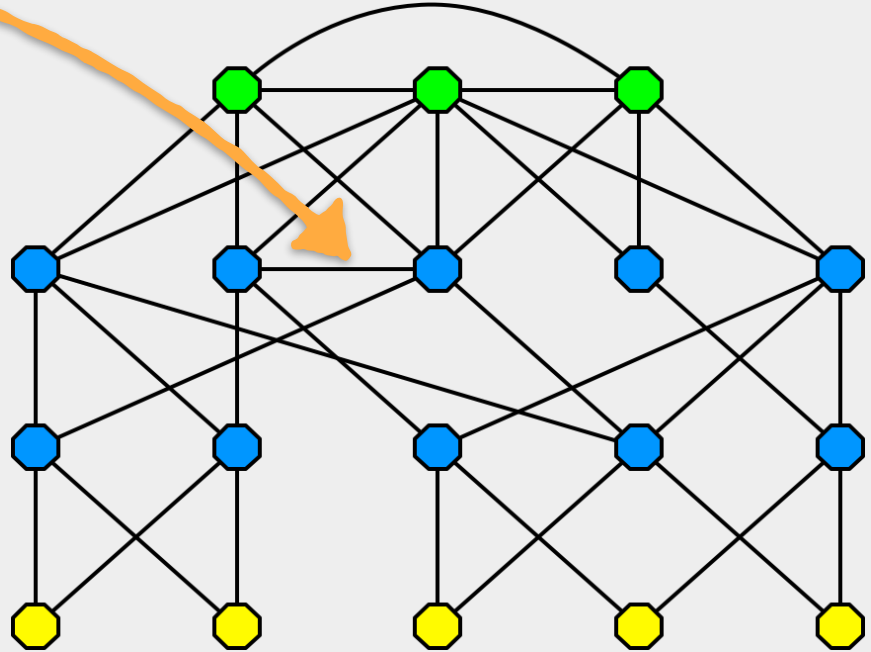
- **Оператори класифіковані по рівням (Tier) згідно з бізнес-моделлю**
  - **Tier 1: нікому не сплачує за трафік**
  - **Tier 2: сплачує тім що вище, бере гроші з тих хто під ним**
  - **Tier 3: не отримує грошей від інших операторів**



А зараз що?

Плюс пірінг

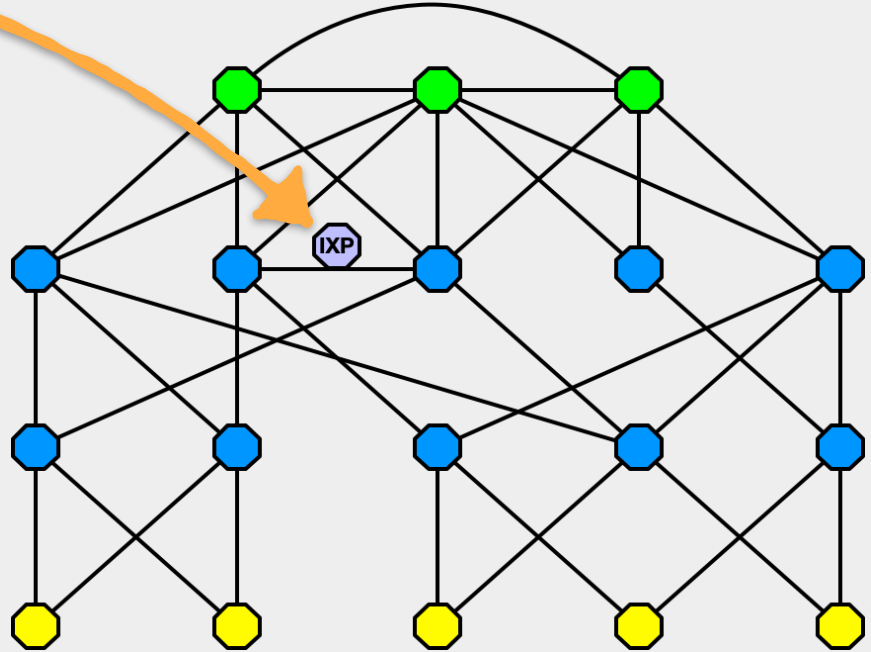
- **Пірінг: обмін власним трафіком та трафіком клієнтів**
- Відверто кажучі, він вже був на діаграмі
  - Між операторами Tier 1



А зараз що?

Також IXP

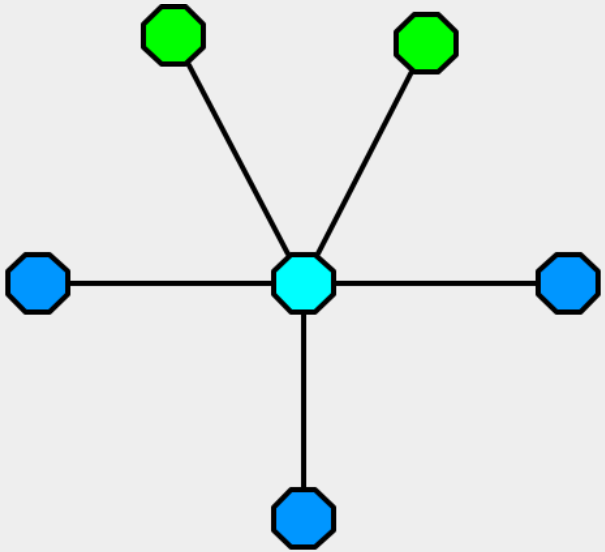
- **Масштабуємо пірінг: точки обміну трафіком (IXPs)**
- **Зазвичай об’єм трафіка не рахується до сплати**
- **IXP мають власні ASN, але вони не додаються до AS PATH**



Світ Гао-Рексфорда

Як мають розповсюджуватися анонси?

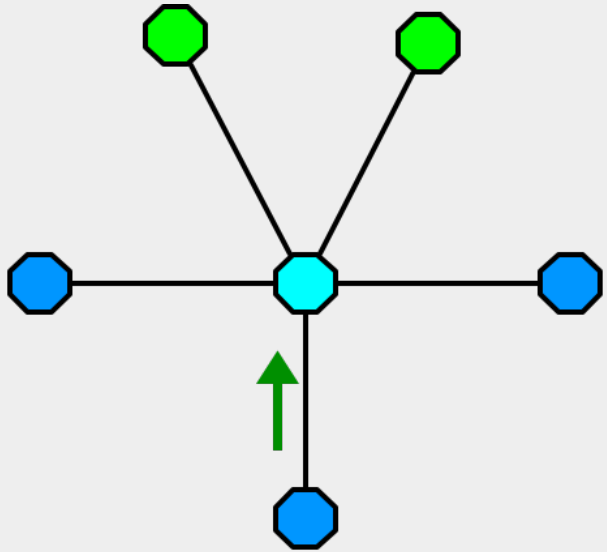
- **Ми у центрі**



Світ Гао-Рексфорда

З тих хто під нами

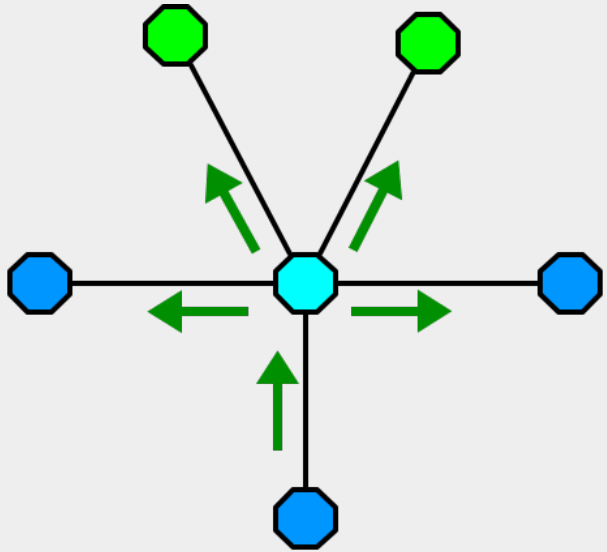
- **Анонс від клієнта (вони нам платять!)**



Світ Гао-Рексфорда

З тих хто під нами - усім іншим!

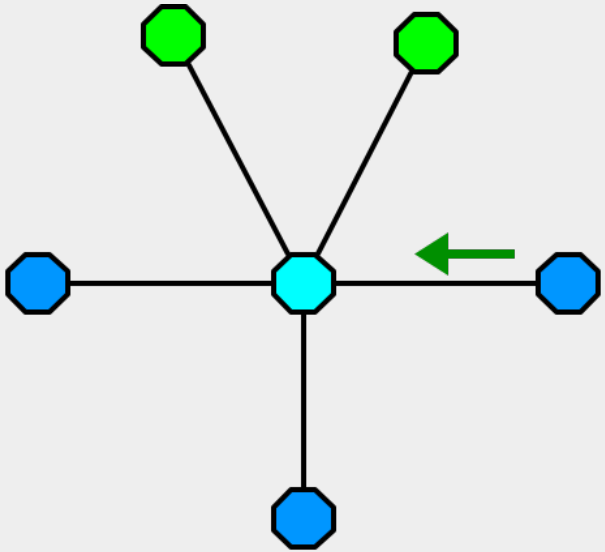
- **Анонс від клієнта (вони нам платять!)**
- **Роздаємо його скрізь!**



Світ Гао-Рексфорда

Від партнера по пірінгу

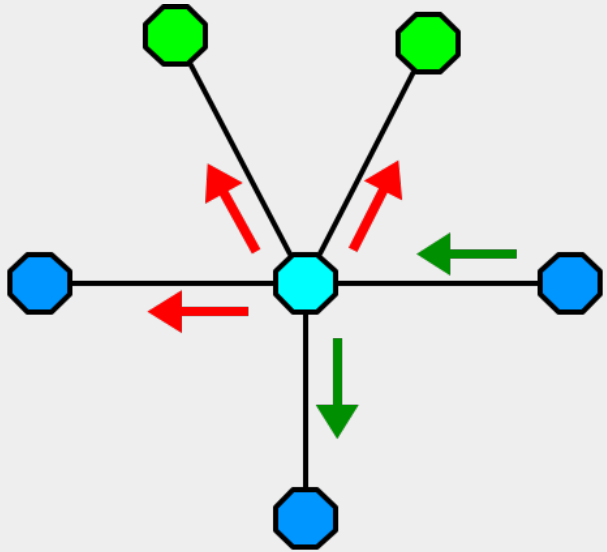
- **Партнер не платить нам за трафік**



Світ Гао-Рексфорда

Від партнера по пірінгу - тільки донизу

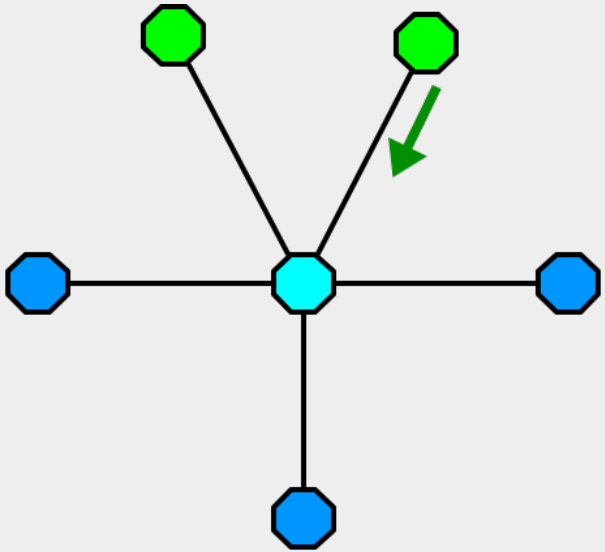
- **Партнер не платить нам за трафік**
- **Анонсуємо його тільки клієнтам (вони сплачуватимуть)**



Світ Гао-Рексфорда

Від аплінка

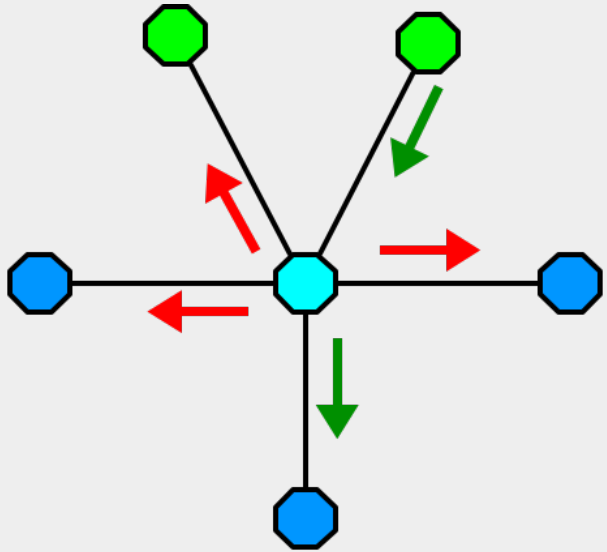
- **Ми платимо аплінку**



Світ Гао-Рексфорда

Від аплінка - тільки клієнтам

- Ми платимо аплінку
- Анонсуємо його тільки клієнтам (вони сплачуватимуть)



Світ Гао-Рексфорда

Коротше

Анонси знизу віддаємо по всім напрямкам

Всі інші анонси віддаємо тільки вниз

Світ Гао-Рексфорда

Коротше

Анонси знизу віддаємо по всім напрямкам

Всі інші анонси віддаємо тільки вниз

АЛЕ!

У BGP нема низу та верху

Це, здається, повна зрада!

Ні, це не вона

Технології вдосконалюються

Багато років індустрія шукала рішення

Більшість з них виявилася неробочими

Але не всі!

Покращення

Що треба пам’ятати

Джерело довіри має бути зовнішнім до системи (у нашому випадку - до роботи протокола BGP)

Треба враховувати бізнес-відносини

Має підтримуватися покрокове впровадження
Обов’язково підтримувати сумісність з попередніми реалізаціями

Конструктор

Не замкнутий механізм, але централізований конструктор з можливостями для розвитку

Можна додавати нові цеглі

Зараз маємо дві: ROA та ASPA

Зовнішній по відношенню до BGP

Жодного нового атрибута чи NLRI

Авторизація обов’язкова, і обійти її неможливо

Resource Private Keys Infrastructure (RPKI)

Дві сторони однієї монетки

Криптографічно зв’язує різні істоти друг до друга
Наприклад, префікс IP та анонсуючу його AS

Побудований на ланцюжках довіри
Якорі (trust anchors) - у RIRів

Тому завжди має дві частини
створення, підпис та публікація “заяв” власниками (**signing**),
перевірка анонсів всіма іншими, згідно з отриманих заяв (**validating**)

Ролі у BGP

Спочатку наведемо дім до ладу

Локально у себе класифікуємо всі свої BGP-сесії
Тут саме цей крок є джерелом правди

Вмикаємо підтримку BGP Roles
Обладнання починає виконувати правіла Гао-Рексфорда

Новий атрібут BGP (має назву OTC)

BGPSEC

Перемога?

BGPsec_PATH замість AS PATH

Ланцюжковий підпис у BGPsec_PATH

Сертифікати рутеров для BGPsec
(з репозиторія RPKI)

Єдина верифікація як джерела анонса (замість ROA) так і шляху передачи (замість ASPA)

(old good)
RPKI ROA

Route Origin Authorisation у RPKI (RPKI ROA)

Побороти крадіжку префіксом

Вже дуже популярна технологія, рівень впровадження у світі дуже високий

Криптографічно **зв’язуємо префікс IP**, на котрий у нас є права використання, **з автономною системою**, котра через це отримає права на те, щоб його анонсувати

Такий файл-зв’язок має назву ROA
Після сворювання він публікується у репозиторії (це фаза **signing**)

Мережа, що виконує перевірку, завантажує собі всі ROA яки є у світі, фільтрує ті що не проходять перевірку, а залишок починає використовувати для перевірки і фільтрації BGP-анонсів, котрі вона отримує від інших мереж (**validating)**

Route Origin Authorisation у RPKI (RPKI ROA)

Де можна дізнатися більше?

Навчальні курси RIPE NCC

- Deploying RPKI Webinar
- Introduction to RPKI Webinar
- BGP Routing Security Training Course (face-to-face, 1 day)
  ! В Україні офіційно - на жаль, тільки після війни

Курси для самостійного навчання: RIPE Academy

- Microlearning course: Why RPKI?
  https://academy.ripe.net/mod/scorm/player.php?a=110&currentorg=articulate_rise&scoid=220
- Microlearning course: What is RPKI?
  https://academy.ripe.net/mod/scorm/player.php?a=116&currentorg=articulate_rise&scoid=232
- Full e-Learning course: BGP Security
  https://academy.ripe.net/course/view.php?id=15

BGP Roles

BGP Roles

“Давайте все це розмалюємо!”

- Для кожної сесії ми вибираємо **певну роль з нашого боку**:
  - customer
  - provider
  - peer
  - RS
  - RS-client

- Вона має відповідати **ролі з боку нашого сусіда**!

- Ми перевіряємо атрибут **OTC (Only-To-Customer) attribut,** котрий має містити “найвищій ASN”
  - Transitive optional
  - Attribute Type Code 35

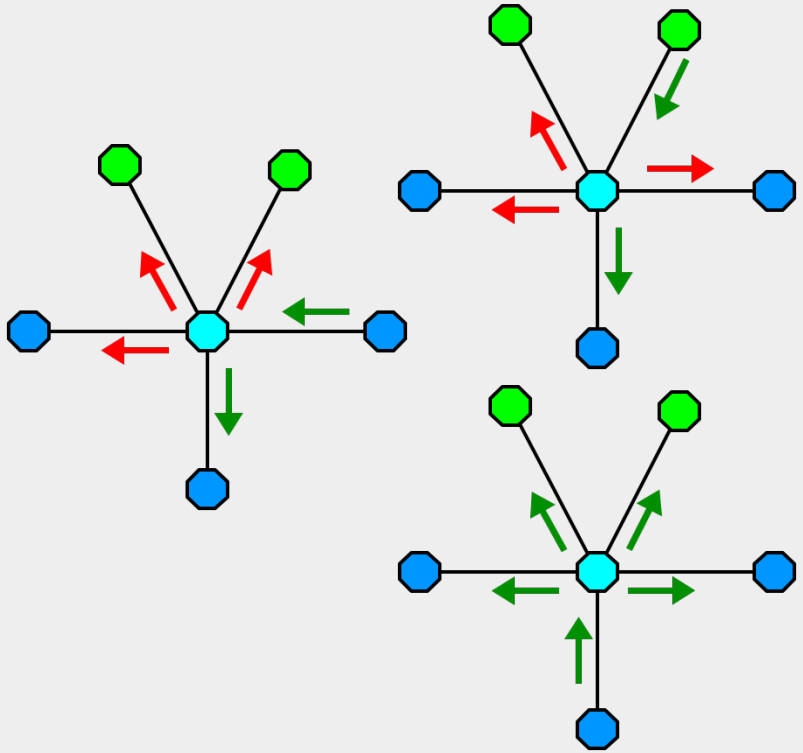
- **Анонс без атрибута OTC, що йде вниз або вбік, має його отримати від нас**

- Анонс **не від клієнта** (особливо з **OTC**) може надсилатися тільки через сесію де ми **provider**

BGP Roles

Вже знайомі діаграми

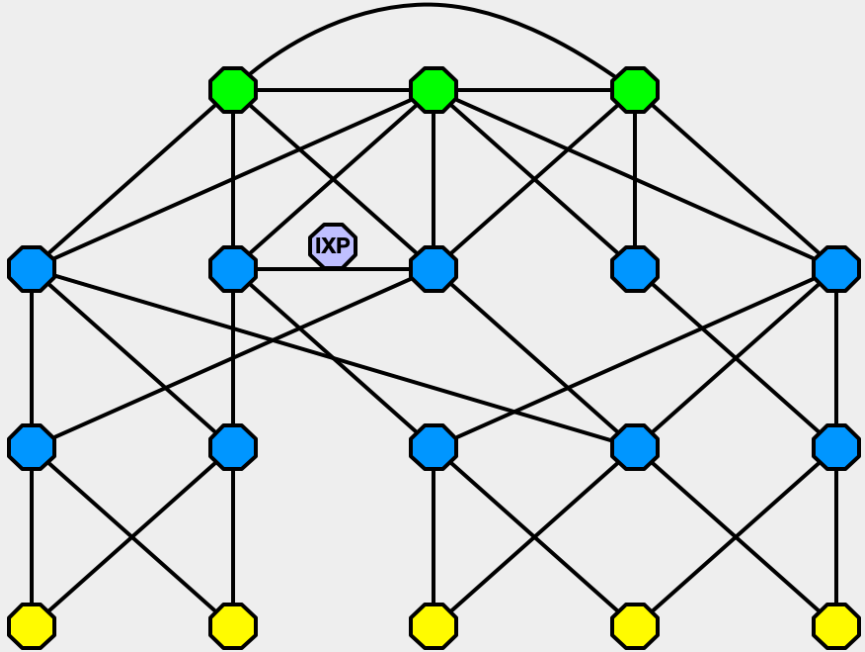
- Якщо немає OTC
  - Просто користуємось правилами Гао-Рексфорда



BGP Roles

Вже знайомі діаграми

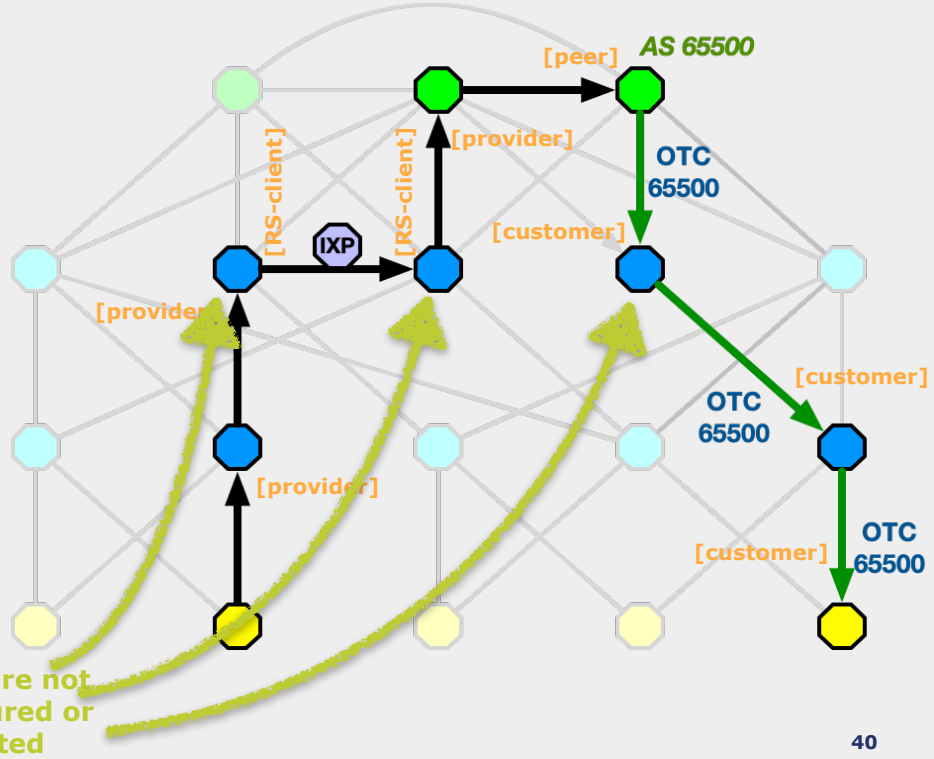
- Якщо є OTC



BGP Roles

Вже знайомі діаграми

- Якщо є OTC
  - Неважливо як ми йдемо у гору
  - Але вниз ми йдемо тільки якщо ми provider
  - Через те що OTC транзитивний, навіть якщо на певному кроці ця логіка не підтримується, вона буде підхоплена далі



BGP Roles

Ну і хто це підтримує?

Софт

BIRD
FRRouting
OpenBGPD
VyOS

Залізо

Juniper
Huawei
MikroTik

RFC 9234 - питайте свого постачальника!

RPKI ASPA

Autonomous System Provider Authorization (RPKI ASPA)

Загальні принципи

Криптографічно **зв’язуємо наш ASN з ASN наших аплінків**
Так ми маємо централізоване сховище бізнес-відносин у Інтернеті

Такий файл-зв’язок має назву **ASPA**; Після сворювання він публікується у репозиторії (це фаза **signing**)

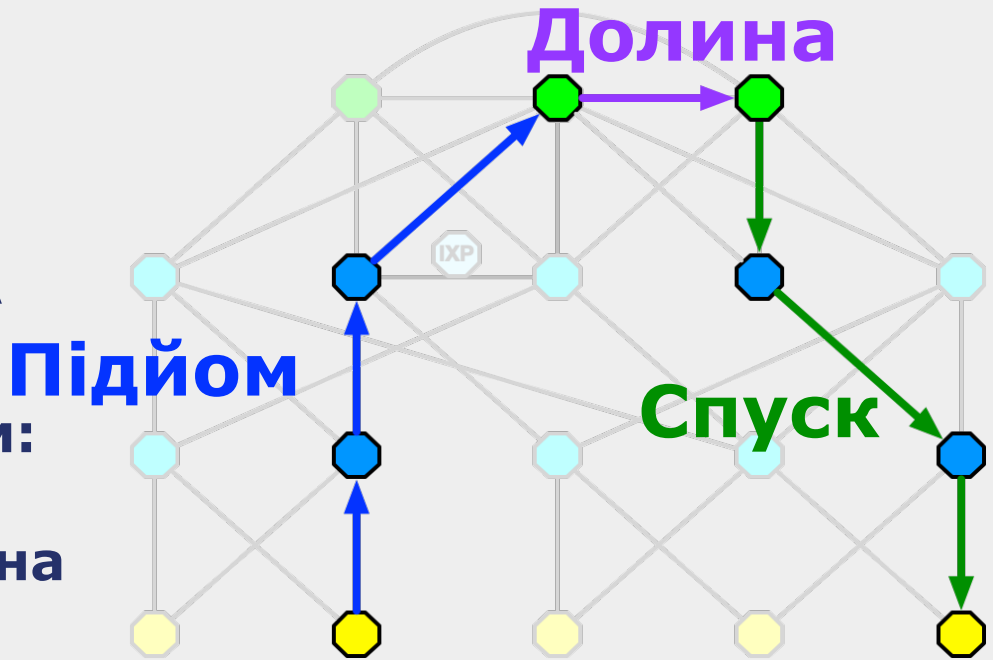
Мережа, що виконує перевірку, завантажує собі всі ASPA яки є у світі, фільтрує ті що не проходять перевірку, а залишок починає використовувати для перевірки AS PATH і фільтрації BGP-анонсів, котрі вона отримує від інших мереж (**validating)**

Autonomous System Provider Authorization (RPKI ASPA)

Правило ASPA: ні долинам!

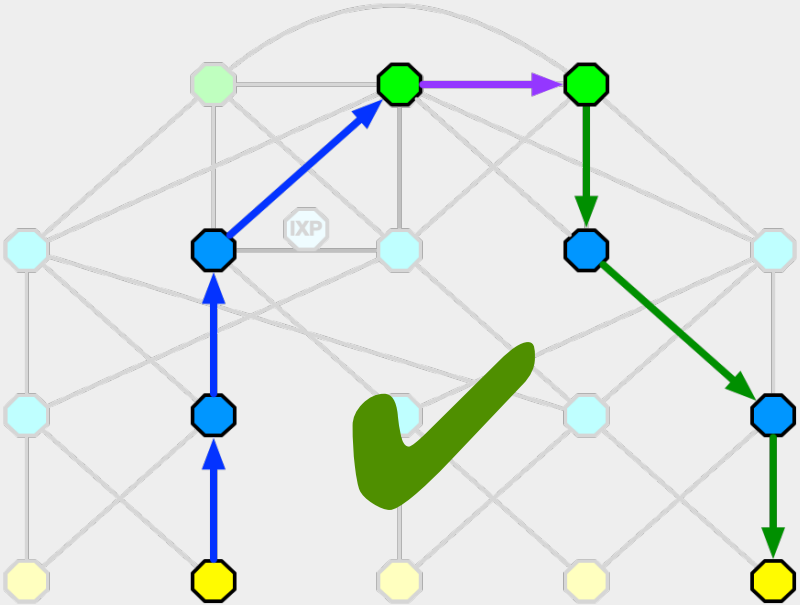
- **Підйом:** сегмент **c ➤ p**
  Спуск: сегмент **p ➤ c**
  Долина: сегмент, не описаний у жодному ASPA

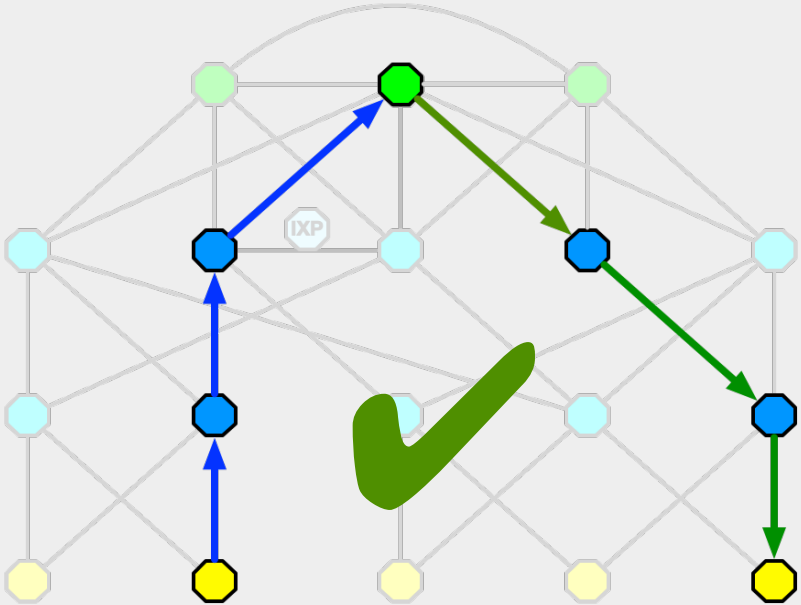
- **AS PATH має бути таким:**
  - **спочатку підйоми**
  - **максимум одна долина**
  - **далі тільки спуски**



Autonomous System Provider Authorization (RPKI ASPA)

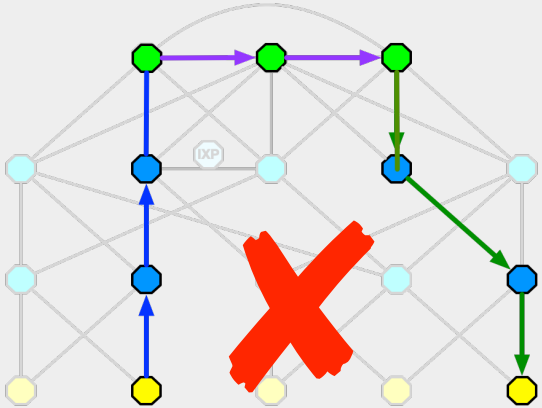
Мабуть, простіше показати, ось як треба:

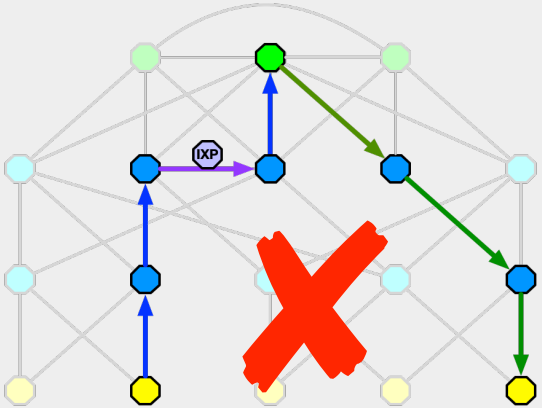


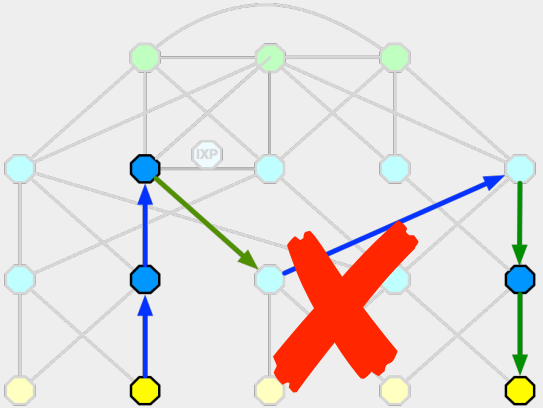


Autonomous System Provider Authorization (RPKI ASPA)

А ось як НЕ треба:







Autonomous System Provider Authorization (RPKI ASPA)

Хто підтримує

RIR Repositories
Production: RIPE NCC, ARIN
Planned: APNIC, LACNIC, AfriNIC

Хто вже користується
RIPE NCC
Cloudflare
Probably Hurricane Electric
Maybe Cogent

Validators
Routinator
FORT Validator
rpki-client

Software routers
OpenBGPD
NIST BGP-SRx

Hardware routers
None at the moment

BGP Roles та ASPA: що може піти не так?

Частковий контроль

BGP Roles

Контролюємо тільки шлях вниз

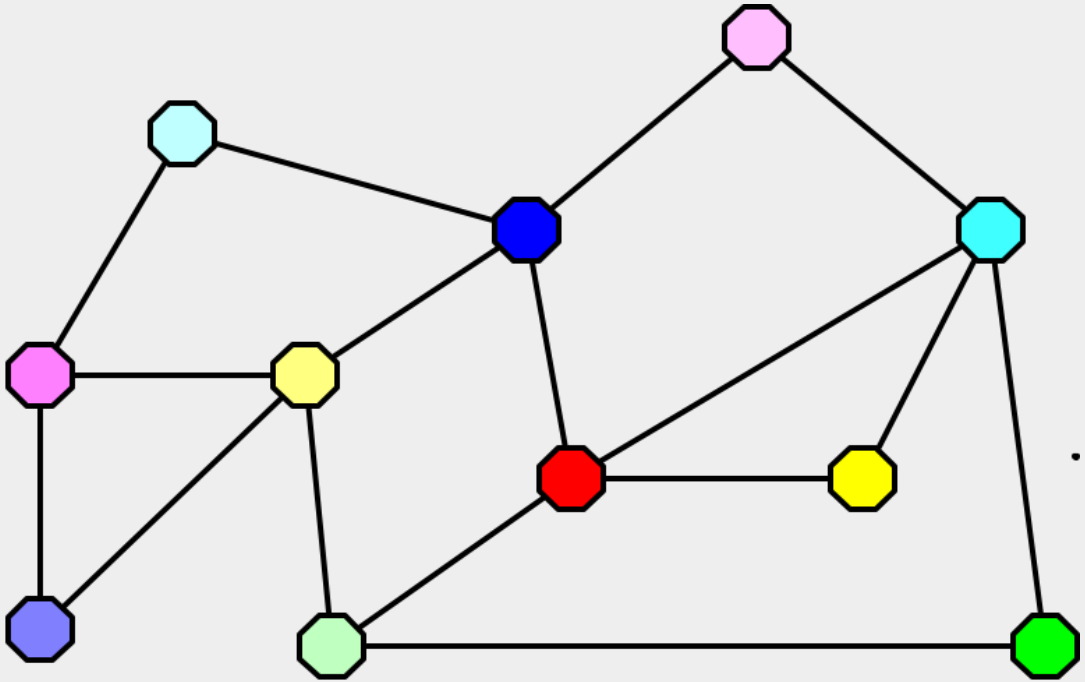
ASPA

Складно повноцінно перевіряти AS PATH коли немає певних ASPA

(“це долина чи відсутня ASPA для c ➤ p?”)

Повернемось на кілько кроків (слайди 12 та 13)

BGP = “гроші, котрі переклали на мову маршрутизаторів”



Складі сценарії (життя бентежне)

Життя бентежне

Оператори можуть мати різні ролі один для одного:

- Для різного трафіку
  (“ти мій провайдер у IPv4, я твій у IPv6”)
- Для різних геолокацій
  (“Я твій провайдер у Азії, ти мій у Європі”)

BGP Roles

- Може бути потрібно декілька сесій для різних типів трафіку

ASPA

- Все гірше, треба заводити різні ASN та додавати трохи чаклунства (типу трансляцій ASN у AS PATH)

BGPSEC

Новий атрибут

AS_PATH більше не є обов’язковим: BGPsec_PATH замість нього
Але не обидва одночасно!

Або BGPsec_PATH, або AS_PATH має бути присутнім

BGPsec_PATH - non-transitive
Якщо наступний рутер не підтримує BGPsec_PATH,
він МУСИТЬ бути зконвертован у AS_PATH
(а зворотня трансформація не існує!)

У BGPsec_PATH є два ланцюжка
Secure_Path: еквівалент традиційного AS_PATH
Signature_Block: послідовність підписов для сегментов Secure_Path

Підписання: технології блокчейну у світі маршрутизації

Кожен маршрутизатор підписує весь Secure_Path та весь попередній Signature_Block

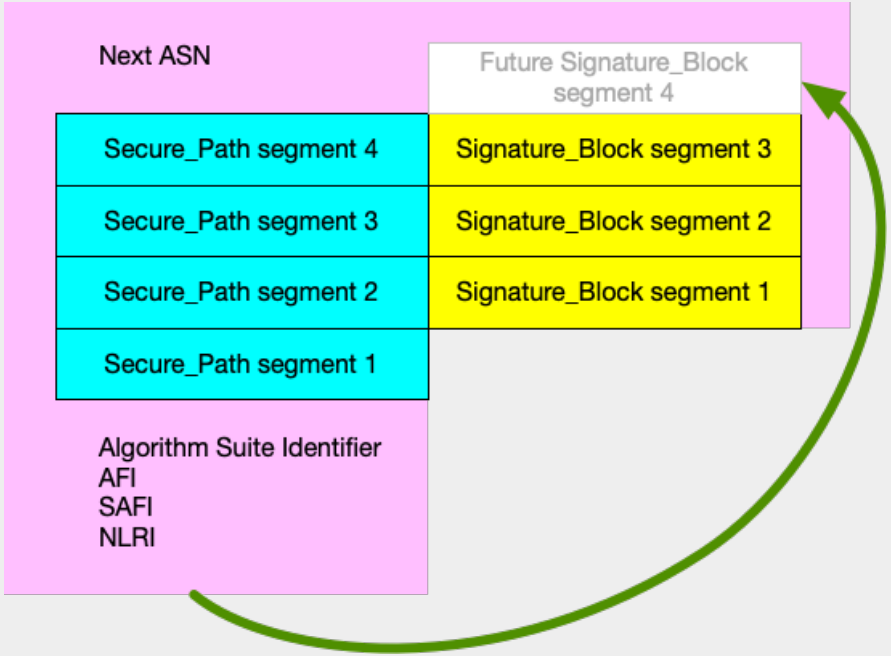
- тобто, для у підпис входять всі попередні підписи

На стадії перевірки ми перевіряємо всі підписи у Signature_Block у зворотному порядку

Для перевірки використовуються BGPsec Router Certificates

- вони мають бути опублікованими у репозиторіях RPKI

Наочно



На кожном кроці створювання підпису додаємо NLRI

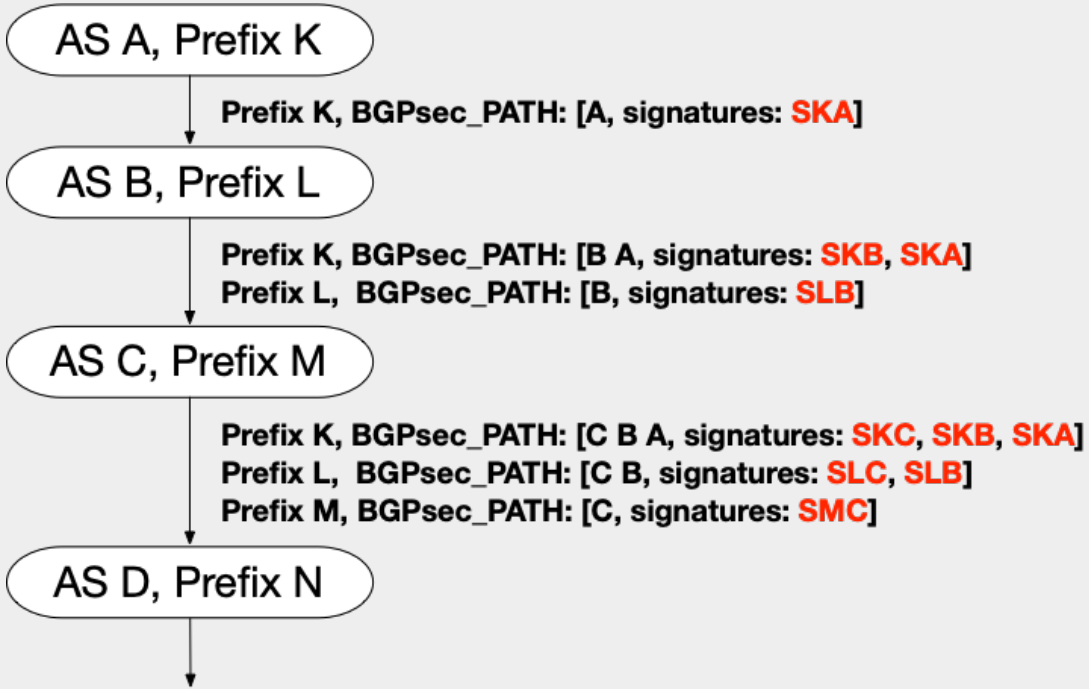
На кожном кроці створювання підпису використовуємо всі попередні підписи

Не дуже багато що можна тримати у кеші

Криптографічні операції неможливо зробити заздалегідь

Перевірка

- Тобто, вся криптографія має бути виконана після отримання анонса та перед його обробкою
- З ростом мережі кількість “важких” операцій зростає **нелінійно**!



Ку-ку!

А ви бачили як у BGPSEC описані бізнес-відносини?

BGPSEC

Ку-ку!

А ви бачили як у BGPSEC описані бізнес-відносини?

Ні, тому що вони ніяк не описані

Поточний стандарт не має засобів боротьби з route leaks

Проблеми

Дуже дорогий с точки зору обчислень

Щоб він приносив користь, має бути близько 100% впровадження

Вирішує тільки частину проблем
Безпека поперед усе, але з іншими проблемами також треба щось робити

Що ще додали до BGP? (список неповний)

Лоскутна ковдра BGP

BGPv4: 1994, IPv6: 1996

Класичний BGPv4: IPv4 NLRI + його атрибути

З появою IPv6 треба було розширити BGP так, щоб він почав працювати не тільки з IPv4

Щоб цього досягнути, був створений MP-BGP: RFC 4760

Лоскутна ковдра BGP

RFC 4760

Multiprotocol Extensions for BGP-4

Замість NLRI у ньому були введені **нові optional, non-transitive атрибути**:

MP_REACH_NLRI (type 14) = AFI+SAFI+NEXT_HOP+NLRI (+ Reserved)
MP_UNREACH_NLRI (type 15) = AFI+SAFI+NLRI

Тобто, не-IPv4 UPDATE тепер містить тільки атрибути

Лоскутна ковдра BGP

RFC 4760

Multiprotocol Extensions for BGP-4

Замість NLRI у ньому були введені **нові optional, non-transitive атрибути**:

MP_REACH_NLRI (type 14) = AFI+SAFI+NEXT_HOP+NLRI (+ Reserved)
MP_UNREACH_NLRI (type 15) = AFI+SAFI+NLRI

Тобто, не-IPv4 UPDATE тепер містить тільки атрибути

Можна створювати будь-які нові NLRI

Лоскутна ковдра BGP

Нові NLRI

Unicast IPv4/IPv6

Multicast IPv4/IPv6

VPNv4/VPNv6

L2VPN та EVPN

Link States (BGP-LS)

Labeled Unicast (BGP-LU)

Route Target Constrains

FlowSpec

…

Лоскутна ковдра BGP

Нові NLRI

Всі вони - частина атрибуту MP_REACH_NLRI

MP_REACH_NLRI не транзитивний, тому якщо BGP-peer його не розуміє, далі він не йде!

Community

Атрибут community: механізм довільного тегування анонсів

Існують різні типи (Basic Community, Extended community, Large Community, IPv6 Address Specific Extended Community)

Є невеличка кількість стандартних значень для них
Обробка навіть стандартних well-known community -
це не обов’язок, а послуга

Інтерпретація значення цього атрибуту залежить від **домовленостей**

Безпека через BGP: FlowSpec

Фільтрація за допомогою BGP

FlowSpec

Механізм розповсюдження правил фільтрації та обмеження трафіку

Окремий NLRI спеціального типу
Задає правила, ***який*** саме трафік має оброблятися спеціальним чином

Окремі Community (ACTIONS)
Задають ***як саме*** має оброблятися трафік, котрій підпав під правила що задані у NLRI

NLRI

Кожен блок має свій тип (починаючи з 1), котрий задає яка частина заголовка описується (адреса отримувача, адреса відправника, номер порта, довжина пакета etc)

Багато типів можуть містити **оператори ==, <, <=, >, >=, !=** або **бітову маску**

Більшість типів (крім адресних) можуть мати **кілька блоків поспіль**
(кожен блок містить оператор **AND або OR** для попереднього блока)

NLRI

Послідовність блоків: строго **від меншого типу до більшого**

Оператор **AND** завжди має **пріоритет над OR**

Обробка NLRI на стадії
трансляції у інструкції ASIC - у тому ж порядку
(а ось внутрішня обробка у ASIC може бути різною)

Таким чином, **FlowSpec NLRI - це дуже довгий запис змінної довжини, котра може описувати достатньо складні правила фільтрації**

FlowSpec NLRI

Types

Type	Component	Encoding / caveat
1	Destination prefix	<type,len,prefix>
2	Source prefix	<type,len,prefix>
3	IP protocol	Value
4	Port	Source OR destination TCP/UDP port(s) (first fragments only)
5	Destination port	TCP/UDP destination port(s) (first fragments only)
6	Source port	TCP/UDP source port(s) (first fragments only)
7	ICMP type	ICMP only (first fragments only)
8	ICMP code	ICMP only (first fragments only)
9	TCP flags	Bitmask (TCP only; first fragments only)
10	Packet length	IP packet length including IP header
11	DSCP	Bitmask for 6 least significant bits of DS field
12	Fragment	DF, IsF, FF, LF bitmask semantics

FlowSpec NLRI

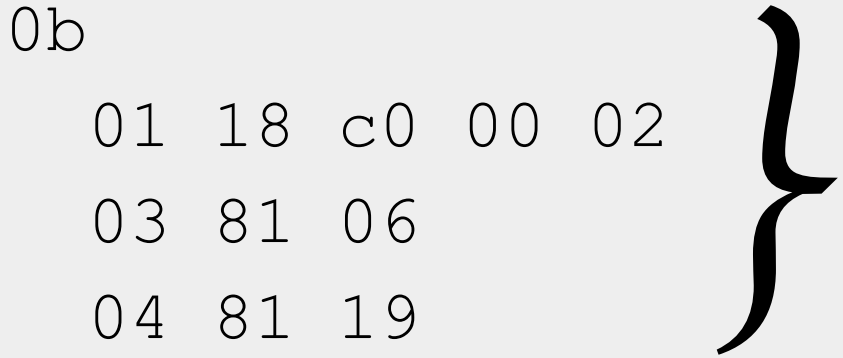
Приклад

ЯКЩО **адреса призначення** є з 192.0.2.0/25,
ТА **протокол** є TCP,
ТА **порт** дорівнює 25,
ТО ...

Length	Type 1 dst prefix	payload	Type 3 protocol	[op,value]+	Type 4 port	[op,value]+	...

FlowSpec NLRI

Приклад




```
0b              NLRI value length = 11 octets

01              component type 1: Destination Prefix
18              prefix length = 24
c0 00 02        192.0.2.0/24

03              component type 3: Protocol
81              end-of-list, length=1, operator ==
06              TCP

04              component type 4: Port
81              end-of-list, length=1, operator ==
19              25
```

Інші атрибути FlowSpec

Реалії BGP

Згідно з RFC4760, UPDATE з MP_REACH_NLRI має містити ORIGIN, AS_PATH и (для iBGP) LOCAL_PREF

ORIGIN: як завжди
(0=IGP, 1=EGP, 2=INCOMPLETE)

AS_PATH: теж як завжди
(на практиці: пустий для iBGP, с Origin ASN для eBGP)

LOCAL_PREF: як і у класичному BGP,
приймає участь у Path Selection Process

Захист NLRI

Перевірка AS_PATH у FlowSpec: FlowSpec validation

У eBGP у відправника мають бути прав на FlowSpec

Тому для FlowSpec UPDATE завжди перевіряється останній (сама ліва) ASN

Він має дорівнювати останньому (самому лівому) ASN
у відповідній BGP RIB для охоплюваючего префікса
(*та вочевидь він взагалі там має бути*)

Тобто, правила фільтрації трафіку може задавати тільки той хто є найкращим маршрутом для цього трафіку

У певних випадках має бути відключена

Конкуренція FlowSpec UPDATE

Роль LOCAL_PREF

Якщо є декілька повідомлень з FlowSpec UPDATE з **ідентичним NLRI**, відбувається Path Selection Process, де приймає участь LOCAL_PREF

(Увага! *NLRI мають бути **ідентичними**, тобто NLRI1=3FFF::/16 та NLRI2=3FFF::/16+протокол(від 0 до 65535) - це геть різні NLRI!*)

Наприклад, контролер А надсилає правило з ACTION=redirect-to-VRF (для очистки),
а контролер В - правило з ACTION=drop (але з гіршим LOCAL_PREF)
Поки контролер А живий, трафік буде відправлятися на очистку

Декілька NLRI у одному UPDATE

Економимо повідомлення

Один UPDATE може містити кілька NLRI, у цьому випадку вони всі незалежно отримують однакові атрибути з цього Update

У тому числі - **рецепти обробки будуть для них співпадати**

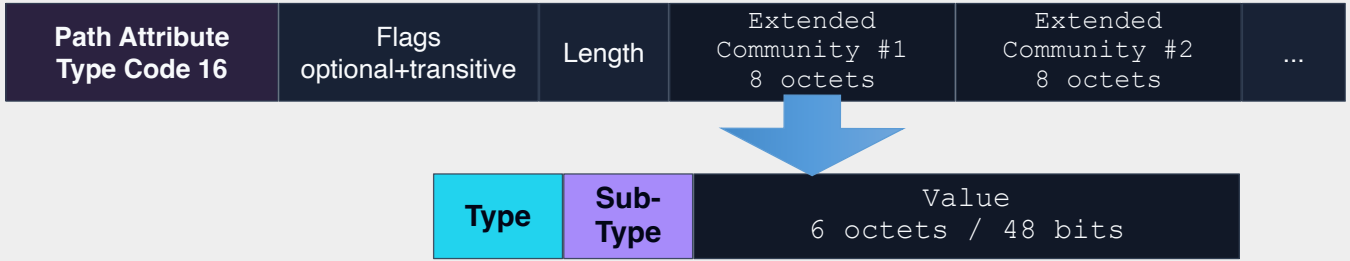
Вказівки по обробці трафіку

ACTIONS

Actions знаходятся у іншому **optional transitive** атрибуті:

```
EXTENDED_COMMUNITIES
  Path Attribute Type Code = 16
  optional transitive
```

Типова Extended Community займає 8 октетів
(можуть бути і більш довгі варіанти)



Кількість ACTIONS

Скільки їх може бути?

Один UPDATE може містити кілька ACTION, у цьому випадку інтерпретація “рецепта” віддана на суд виробника обладнання (implementation-specific)

Також, **у UPDATE може взагалі не бути жодного ACTION**, тоді виконується дія за замовчуванням: дозволити трафік (можна використовувати на етапі підготовки і перевірки)

Що можна робити?

Code	Action	Value payload	Notes
0x8006	traffic-rate-bytes	2-octet info + 4-octet IEEE float bytes/s	0 = discard
0x800c	traffic-rate-packets	2-octet info + 4-octet IEEE float packets/s	conflicts with bytes-rate
0x8007	traffic-action	6-octet bit field; defined bits: 47=T, 46=S	terminal/sample
0x8008	rt-redirect AS2	2-octet AS + 4-octet local admin	redirect-to-VRF RT
0x8108	rt-redirect IPv4	IPv4 address + 2-octet local admin	redirect-to-VRF RT
0x8208	rt-redirect AS4	4-octet AS + 2-octet local admin	redirect-to-VRF RT
0x8009	traffic-marking	DSCP in 6 least significant bits	remark DSCP

ACTION

traffic-rate-bytes та traffic-rate-packets

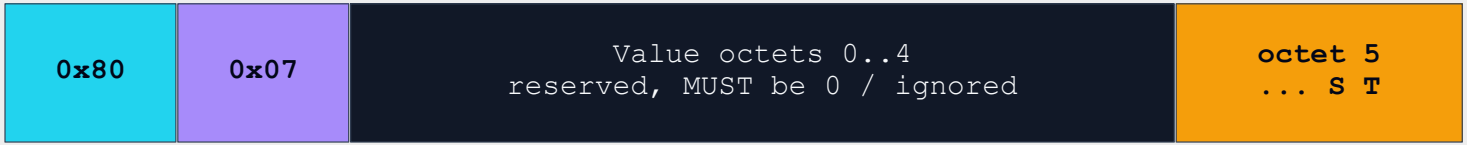
Вказівка щодо **швидкості**, с котрою відправник повідомлення може приймати такий трафік (*треба вказати **або** traffic-rate-bytes у bps **або** traffic-rate-packets у pps, сумісно їх використовувати не можна*)

Швидкість=0 означає, що трафік має бути **відкинуто** (окремої дії DROP не існує)

Механізм боротьби з Volumetric DDoS

ACTION

traffic-action



Мета-запис, наразі мають сенс тільки 2 біти

Біт T
T=1: “продовжуй обробляти наступні FlowSpec записи”
T=0: “після цього запису припини обробку”

Біт S
S=1 вмикає семплювання чи логування цього трафіку

Потрібна підтримка на обладнанні

ACTION

redirect-to-* через Route Target

Code	RT format	Payload	Коли зручно
0x8008	Two-octet AS specific	2B AS + 4B local admin	AS <= 65535 / legacy RT
0x8108	IPv4 address specific	4B IPv4 + 2B local admin	RT based on router/operator address
0x8208	Four-octet AS specific	4B AS + 2B local admin	4-byte ASN operators

Це не маршрут, FlowSpec не задає шлях доставки!

Трафік тільки змінює “місце життя” на відповідний VRF, а далі форвадиться там повністю за правилами того VRF

Зручний спосіб для направлення трафіку на очищення
(до scrubbing center)

Обробка кількох записів FlowSpec

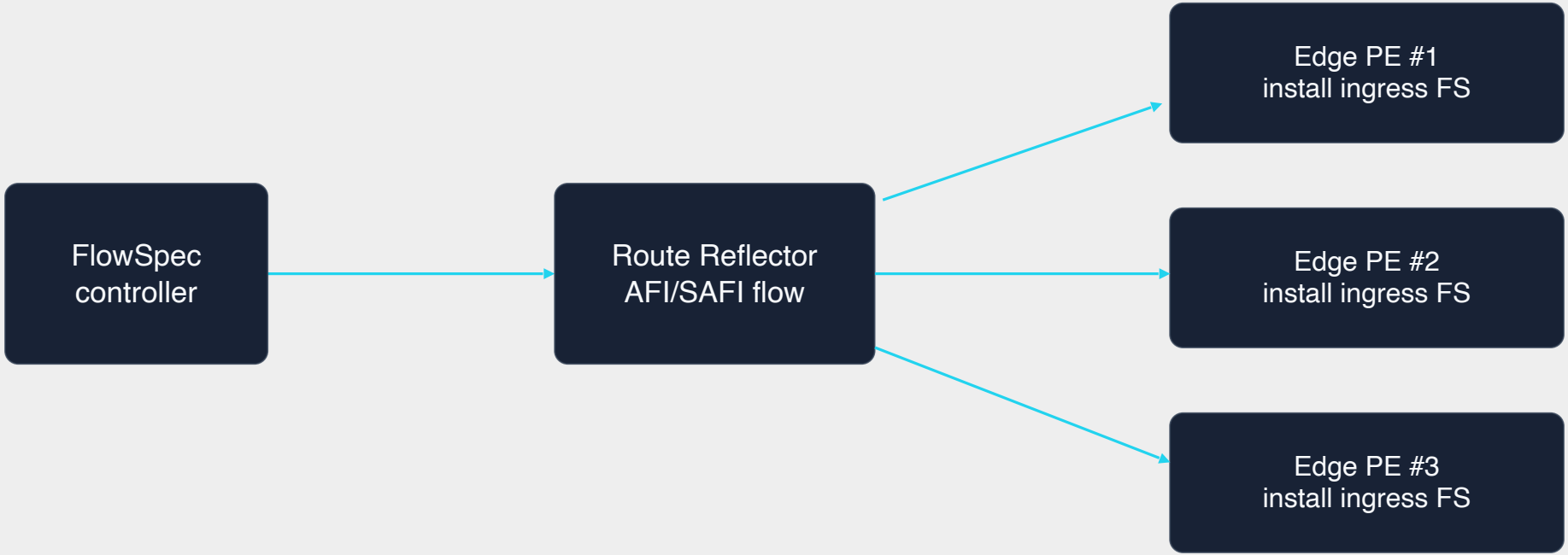
(якщо обладнання підтримує)

Розширена інтерпретація “more specific”:

1. More specific prefix - як завжди
2. Додаткова інформація робить запис more specific:
   Prefix=3FFF::/16+TCP є більш специфічним ніж Prefix=3FFF::/16
3. Додаткова інформація порівнюється за об’емом:
   1 порт більш специфічний за 16 портів
4. Якщо порівнюється додаткова інформація різних типів, більший по номеру тип додає більшу специфіку
5. Бітом T можна задати проходження кількох рецептів поспіль

Робота по iBGP

Типова схема



Робота по iBGP

Контролер FlowSpec

Можна використовувати FlowSpec у своїй мережі
щоб мати розподілений механізм керування фільтрацією

Але у мережі має бути о**дне джерело правди FlowSpec**
(скажімо, на базі GoBGP або ExaBGP)

І **воно не є джерелом анонсів**

Це той випадок, коли валідацію FlowSpec треба відключати: RFC9117
Треба явно визначати Local Domain і хто може інжектити порожній FlowSpec AS_PATH
За межами Local Domain - сувора валідація та/або фільтрація

Не вмикайте Graceful Restart/LLGR без рішення, що застаріли
фільтри мають жити після падіння контролера!

Робота по eBGP

Найбільш типовий випадок

Делегування фільтрації трафіку більш потужному аплінку
(Частіше за все - боротьба з Volumetric DDoS)

Має сенс коли канал від нас до аплінка вже забитий, а канали аплінка мають вільний ресурс

Робота по eBGP

Provider → Customer

Дозволяємо тільки те що **підтримується** нашим обладнанням

Контролюємо **AFI/SAFI**

Контролюємо фільтрацію **керуючого трафіку**
(скажімо, не фільтруємо BGP з клієнтом навіть якщо він хоче)

Контролюємо **ширину** префіксів та діапазонів

Приймаємо **тільки для префіксів клієнта**
(не дозволяємо NLRI тільки з src prefix)

Якщо треба, **коригуємо ACTION**
(наприклад, забороняємо redirect або S=1 у traffic-action)

…

Робота по eBGP

Provider → Provider

Договір з чіткими правилами дозволу для обох операторів

Що можна обробляти кожному

Які ACTION погоджені для кожного

Дії операторів у випадку нештатної ситуації

Робота по eBGP

Customer → Provider

Отримання списку обов’язків від оператора:
Що можна обробляти на його інфраструктурі (вказуя у NLRI)
Що і у якому обсязі підтримується (ACTION тощо)
Як ця послуга тарифікується

Робота по eBGP

Customer → Provider

Отримання списку обов’язків від оператора:

Що можна обробляти на його інфраструктурі
(через FlowSpec NLRI)

Що і у якому обсязі підтримується
(ACTION, max-prefix/max-rules/update-rate, IPv6 EH тощо)

Як ця послуга тарифікується

Що треба пам’ятати

Обмеження

Чи приймаються повідомлення FlowSpec

Чи підтримується глибоке сканування пакета?
(*скажімо, заголовок TCP після Extension Headers*)

Чи підтримується робота з діапазонами портів?

Яку кількість записів FlowSpec дозволяє TCAM?

Як на даному обладнанні співвідносяться QoS/PBR/ACL та FlowSpec?
(*який саме порядок обробки*)

(і ще багато аналогічних питань)

Як запобігти проблем

Більшість інцидентів з FlowSpec — не від незнання формату повідомлення або інших деталей, а від:

-перекриття правил

-нерозуміння обмежень

-невідповідної валідації

-наявності застарілих станів

-некоректної обробки фрагментів

Автоматизація

Зазвичай **ніхто не хоче керувати FlowSpec вручну**
Особливо коли на нього прямо зараз відбувається атака

Тому більшість можливостей FlowSpec ризикує залишатися у теорії

Частіше за все використовується **саморобний FlowSpec-контролер** на базі **GoBGP** або **ExaBGP**, котрі вміють створювати FlowSpec UPDATEs

Популярна альтернатива - продукти класу **FastNetMon**
(аналіз трафіку та автоматична генерація FlowSpec-повідомлень аплінку у разі знаходження паттерна атаки)

Ще трохи дегтю

Що ще може вам заважати

У аплінка може бути недостатня смуга "нагору"
Без неї захист через FlowSpec вам ніяк не допоможе:
DDoS виведе з ладу не тільки вас, але і аплінка

Наразі не існує "ієрархічного FlowSpec"
Ваші фільтри аплінк до свого аплінка надсилати не буде

Аплінк може не надавати послугу обслуговування FlowSpec
- Він може мати обладнання котре не підтримує FlowSpec
- Така послуга може конфліктувати з його бізнес-моделью

Обладнання аплінка може не підтримати потрібну *вам* фільтрацію

Глибокий аналіз не підтримується
Частина трафіку атаки може проходити якщо фільтр занадто вузький
(скажімо, другий та наступні сегменти UDP)

RIPE NCC
RIPE NETWORK COORDINATION CENTER

Questions?
Comments?
Applauses?

asemenyaka@ripe.net