IPv6 and RPKI in Timor-Leste

Jose Dias
TLNOG

TLNOG2 Conference

F-FDTL Auditorio Room, Fatuhada, Dili Timo-Leste
21 November 2025

Securing and Future-Proofing

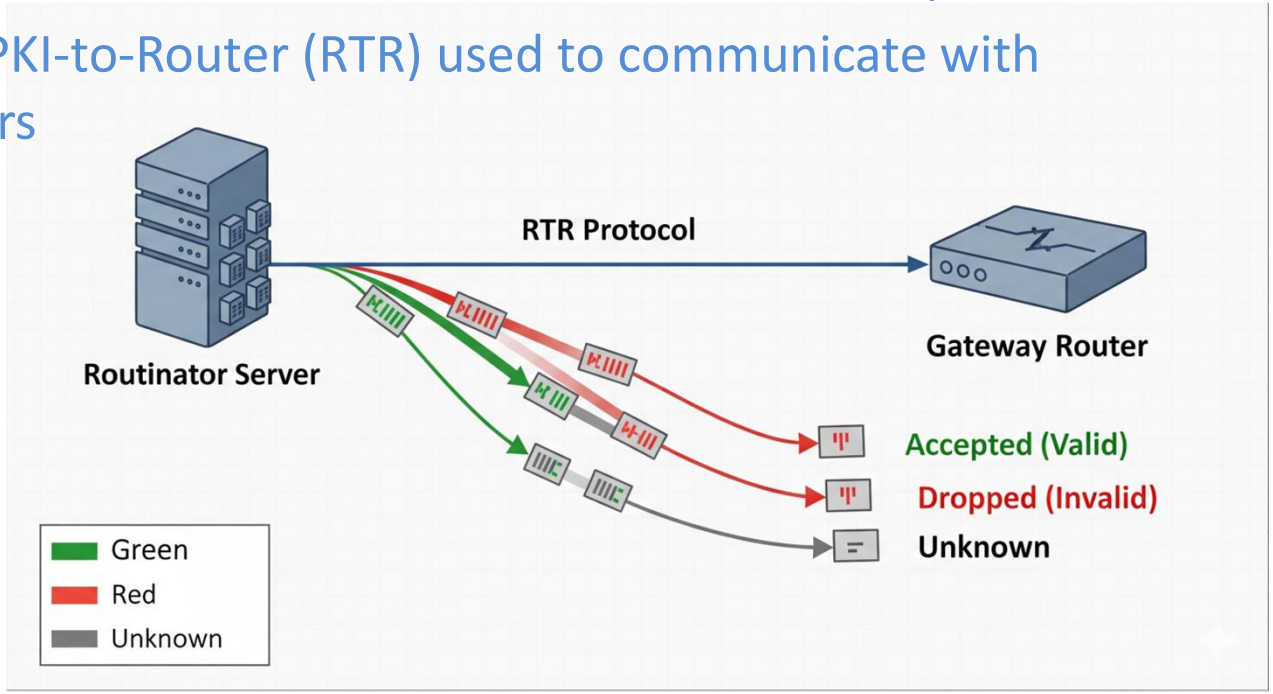
To address IPv4 exhaustion and enhance routing security, some (case study) has implemented IPv6 and Resource Public Key Infrastructure (RPKI). This presentation outlines the practical approach and architecture.

Why RPKI?	**Why IPv6?**
A security framework that prevents route hijacking and other BGP attack by cryptographically verifying the association between an IP address block and an Autonomous System (AS).	Provide virtually limitless pool of IP addresses, ensuring long-term network growth, innovation and connectivity for new devices and services.

RPKI Implementation

Architecture

- Two RPKI cache server deployed for redundancy
- Software: Routinator installed on Ubuntu
- Function: Validate BGP announcements from IP Transit providers.
- Integration: RPKI-to-Router (RTR) used to communicate with gateway routers



RPKI Implementation

Routinator Server (Ubuntu)

- Installation

```
https://github.com/NLnetLabs/routinator
```

Or

```
$ sudo apt install routinator
```

- Configuration (/path/to/routinator.conf)

```
repository-dir = "/var/lib/routinator/rpki-cache"
rtr-listen = ["[your_validator_v4/v6_IP]:8323"]
http-listen = ["your_validator_v4/v6_IP:8423"]
```

- Fire-Up the Routinator

```
$ sudo systemctl start routinator.service
```

RPKI Implementation

Routinator Server (Ubuntu)

- Allowed TCP port for RTR and HTTP-custom-port

```
$ sudo ufw allow 8323/tcp
$ sudo ufw allow 8423/tcp
```

- Verify with netstat

```
$ sudo netstat -antp |grep 8323
$ sudo netstat -antp |grep 8423

#output-ommitted
tcp6  0  0 [your_validator_v4/v6_IP]:8323  :::*      LISTEN   6693/routinator

#output-ommitted
tcp   0  0 [your_validator_v4/v6_IP]:8423  0.0.0.0:* LISTEN   6693/routinator
```

RPKI Implementation

Routinator Client (Cisco IOS)

- Configuration

```
router bgp [your_asn]
bgp rpki server tcp [your_validator-v4/v6_IP] port 8383refresh 120
```

- Validation

```
route-map ROUTE-VALIDATION permit 10
 match rpki valid
 set local-preference 200
!
route-map ROUTE-VALIDATION permit 20
 match rpki not-found
 set local-preference 100
!
route-map ROUTE-VALIDATION deny 30
 match rpki invalid
```

RPKI Implementation

Routinator Client (Cisco IOS)

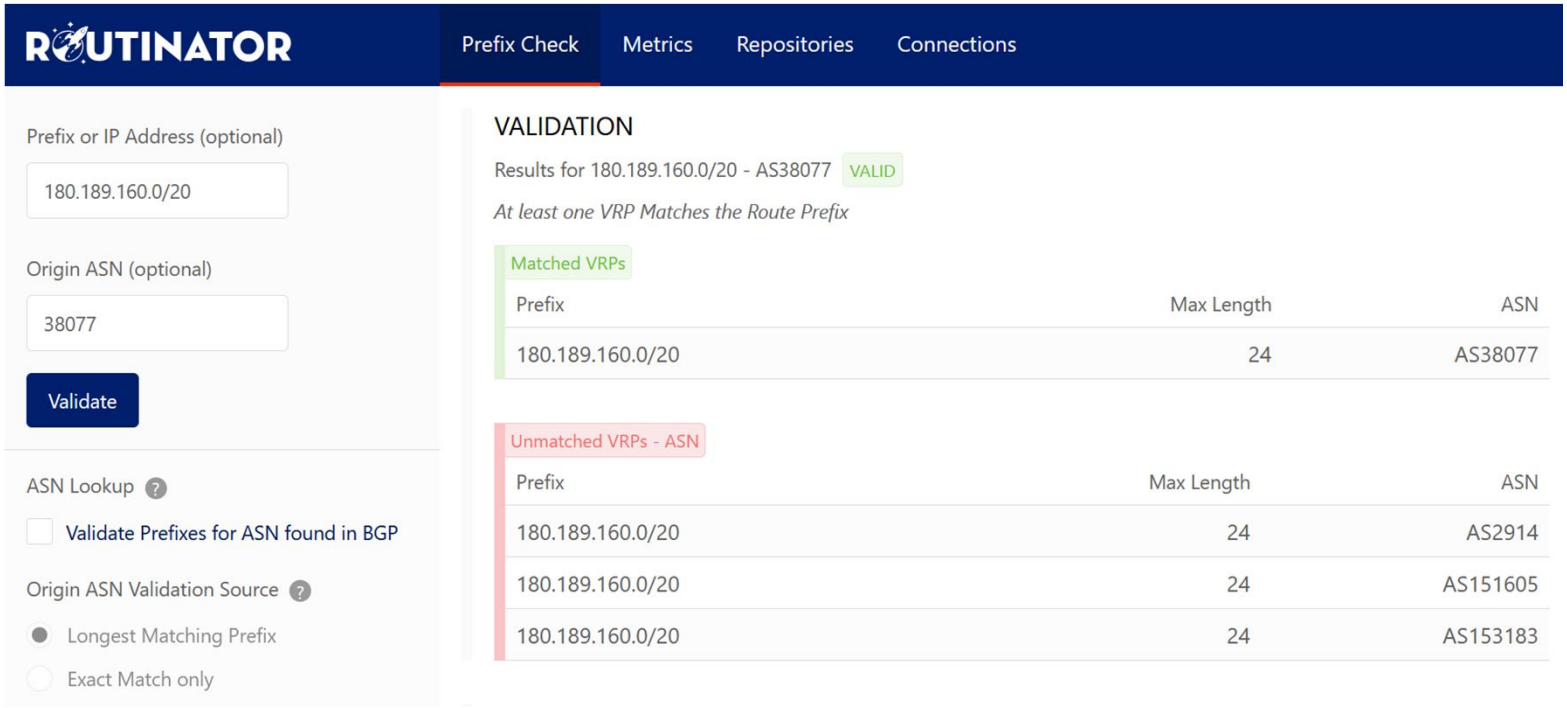
- Apply to the inbound updates

```
router bgp [your_asn]
 address-family ipv4 unicast
  neighbor [your_bgp_peer_v4IP] activate
  neighbor [your_bgp_peer_v4IP] route-map ROUTE-VALIDATION in
 exit-address-family
!
 address-family ipv6 unicast
  neighbor [your_bgp_peer_v6IP] activate
  neighbor [your_bgp_peer_v6IP] route-map ROUTE-VALIDATION in
 exit-address-family
```

RPKI Verification

Routinator Server

– http://[your_validator_v4/v6_IP]:8323



RPKI Verification

Routinator client (Cisco IOS)

- RTR Session

```
#sh bgp ipv4 unicast rpki servers
```

- RPKI Cache

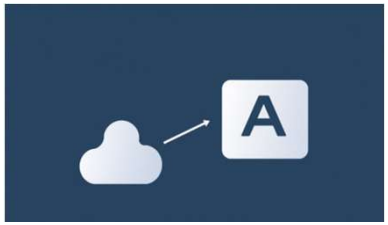
```
#sh bgp ipv4 unicast rpki table
```

- Validation

```
#sh bgp ipv4 unicast 180.189.160.0/20
```

IPv6 Transition: 464XLAT Model

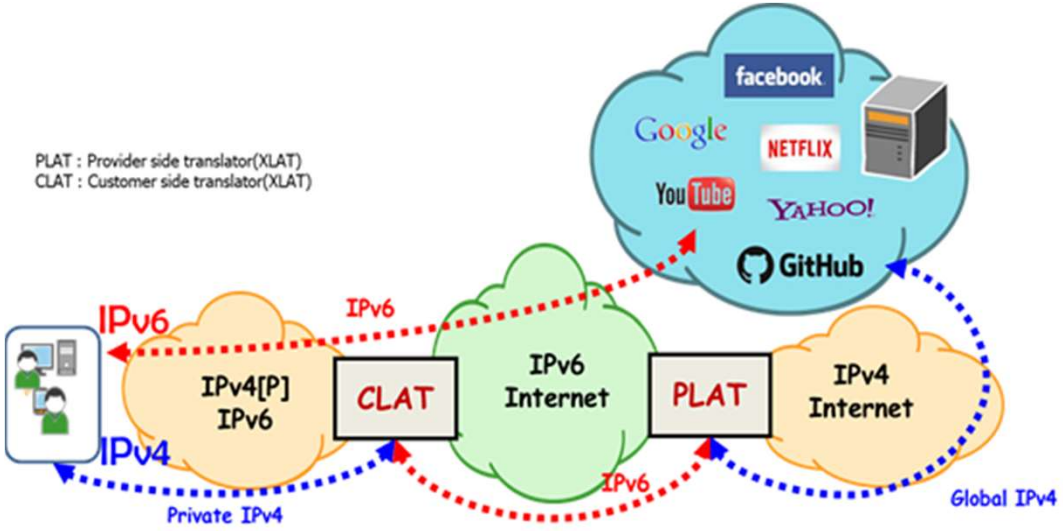
The adoption of 464XLAT transition mechanism is to provide seamless access to IPv4 internet for IPv6-Only clients. It involves a Provider-Side Translation (PLAT) and a Customer-Side Translation (CLAT).

PLAT: NAT64	PLAT: DNS64	CLAT
A cisco router performs stateful NAT between IPv6 and IPv4, allowing the IPv6 clients to reach IPv4 servers.	Ubuntu servers running BIND9 synthesize IPv6 (AAA) records for domains that only have IPv4 (A) records	A MikroTik router with OpenWrt firmware perform stateless translation, enabling IPv4-only applications on client's device to work.
		

464XLAT Implementation

Architecture

- Cisco router as NAT64
- Two DNS64 server deployed for redundancy
- BIND9 installed on Ubuntu
- MikroTik with OpenWrt firmware installed for CLAT on CPE



464XLAT Implementation

PLAT: NAT64 (Cisco IOS)

- Define and allocate v6-only-customer prefixes

```
#ipv6 access-list v6_only_customers-tov4internet
  permit ipv6 2001:db8:8F01:100::/56 any
  permit ipv6 2001:db8:8F00:1::/64 any
```

- NAT64 Configuration

```
(Prefix allocated to be later on adding to BIND9 DNS64)
#nat64 prefix stateful 2001:db8:8FFF::1:0:0/96

(IPv4 pool allocated to v6-only-customer)
#nat64 v4 pool v4_poolfor_ v6-only-customer 192.168.100.100
192.168.100.200
```

464XLAT Implementation

PLAT: NAT64 (Cisco IOS)

- NAT64 configuration

```
(overload the translation allowed v6-only-customer to access IPv4
Internet servers)
#nat64 v6v4 list v6_only_customers-tov4internet pool v4_poolfor_
v6-only-customer overload
```

- IPv6 DHCP prefix delegation

```
#ipv6 local pool Customer_Global_Pool 2001:db8:8F01:100::/56 56

#ipv6 dhcp pool v6Customer-prefix-delegation
  prefix-delegation pool Customer_Global_Pool
  dns-server 2001:db8:100::163
  dns-server 2001:db8:100::164
  domain-name your_domain-name.tl
```

464XLAT Implementation

PLAT: NAT64 (Cisco IOS)

- Apply prefix delegation and NAT64 to interface facing v6Customer

```
#interface GigabitEthernet0/1/4
  description - To v6Customer01
  ipv6 address 2001:db8:8F00:1::FFFE/64
  ipv6 enable
  ipv6 nd other-config-flag
  ipv6 dhcp server v6Customer-prefix-delegation
  nat64 enable
```

- Enable NAT64 on interface facing Internet

```
#interface TenGigabitEthernet0/0/0
  description - To Global Internet
  ipv6 address 2001:db8:1:28::1/127
  nat64 enable
```

464XLAT Implementation

PLAT: DNS64 (BIND9 on Ubuntu)

- Configuration (/path/to/named.conf.options)

```
options { directory "/var/cache/bind"; max-cache-size 200M;

        forwarders { 2606:4700:4700::1111; 2001:4860:4860::8888; };

        //dnssec-validation auto;
        auth-nxdomain no;    # conform to RFC1035
        listen-on-v6 { any; };
        querylog yes;
        recursion yes;
        recursive-clients 1000;
        tcp-clients 1000;

        // Customers - v6 Only Network
        //============================================================
        dns64 2001:db8:8fff::1:0:0/96 {
              clients {2001:db8:8f01:100::/56; 2001:db8:8F00:1::/64;};
              mapped {!10/8; !172.16/12; !192.168/16; any;};
              exclude { 0::/3; 4000::/2; 8000::/1; 2001:db8::/32; };
              break-dnssec no;
              };
};
```

464XLAT Implementation

CLAT (MikroTik with OpenWrt firmware)

- Configuration (/path/to/network)

```
config globals 'globals'
        option ula_prefix 'fd1b:4d53:2f7f::/48'

config interface 'wan6'
        option device 'eth1'
        option proto 'dhcpv6'
        option reqaddress 'try'
        option reqprefix 'auto'

config device
        option name '464-clat'

config interface 'clat'
        option proto '464xlat'
        option tunlink 'wan6'
        option ip6prefix '2001:db8:8FFF::1:0:0/96'
```

464XLAT Verification

CLAT (MikroTik with OpenWrt firmware)

- Pinging 1.1.1.1 and 8.8.8.8

```
root@MikroTip:~# ping 1.1.1.1 -c 3
PING 1.1.1.1 (1.1.1.1): 56 data bytes
64 bytes from 1.1.1.1: seq=0 ttl=54 time=51.628 ms
64 bytes from 1.1.1.1: seq=1 ttl=54 time=53.457 ms
64 bytes from 1.1.1.1: seq=2 ttl=54 time=51.403 ms

--- 1.1.1.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 51.403/51.998/53.457 ms
root@root@MikroTip:~# ping 8.8.8.8 -c 3
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: seq=0 ttl=118 time=58.267 ms
64 bytes from 8.8.8.8: seq=1 ttl=118 time=57.924 ms
64 bytes from 8.8.8.8: seq=2 ttl=118 time=58.002 ms

--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 57.766/57.989/58.267 ms
```

464XLAT Verification

CLAT (MikroTik with OpenWrt firmware)

- DNS query to server with only IPv4 address (www.mj.gov.tl)

```
root@MikroTik:~# nslookup www.mj.gov.tl

Non-authoritative answer:
www.mj.gov.tl   canonical name = mj.gov.tl
Name:   mj.gov.tl
Address: 2001:db8:8fff::1:b4bd:a75d

Non-authoritative answer:
www.mj.gov.tl   canonical name = mj.gov.tl
Name:   mj.gov.tl
Address: 180.189.167.93
```

464XLAT Verification

CLAT (MikroTik with OpenWrt firmware)

- DNS query to server with only IPv4 address (www.tic.gov.tl)

```
root@MikroTik:~# nslookup www.tic.gov.tl

Non-authoritative answer:
www.tic.gov.tl  canonical name = webtic.tic.gov.tl
Name:   webtic.tic.gov.tl
Address: 103.143.164.162

Non-authoritative answer:
www.tic.gov.tl  canonical name = webtic.tic.gov.tl
Name:   webtic.tic.gov.tl
Address: 2001:db8:8fff::1:678f:a4a2
```

464XLAT Verification

PLAT Side

- NAT64 (Cisco OS)

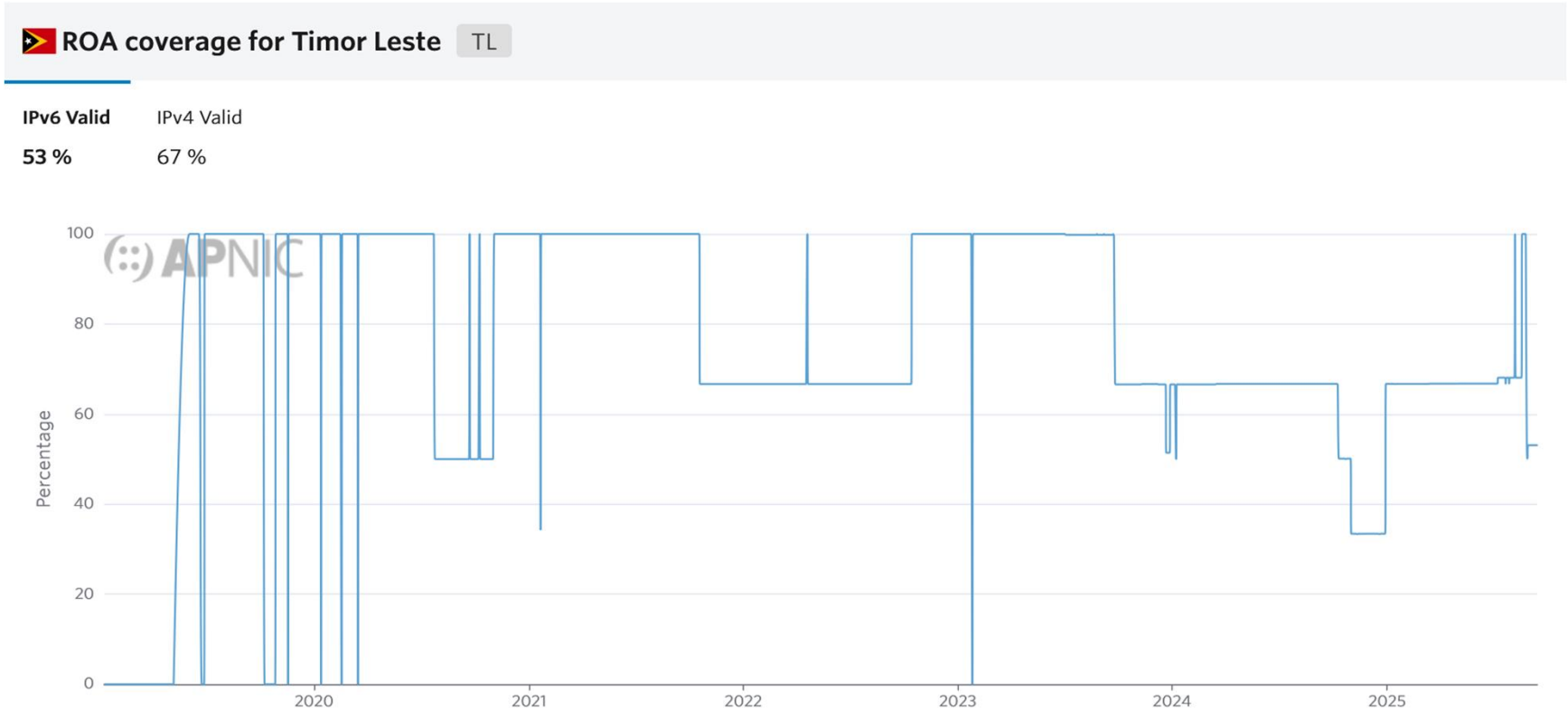
```
(Check NAT64 translation)
#show nat64 translations

(Check IPv6 prefix delegation)
#show ipv6 dhcp binding
```

- DNS64 (BIND9 on Ubuntu)

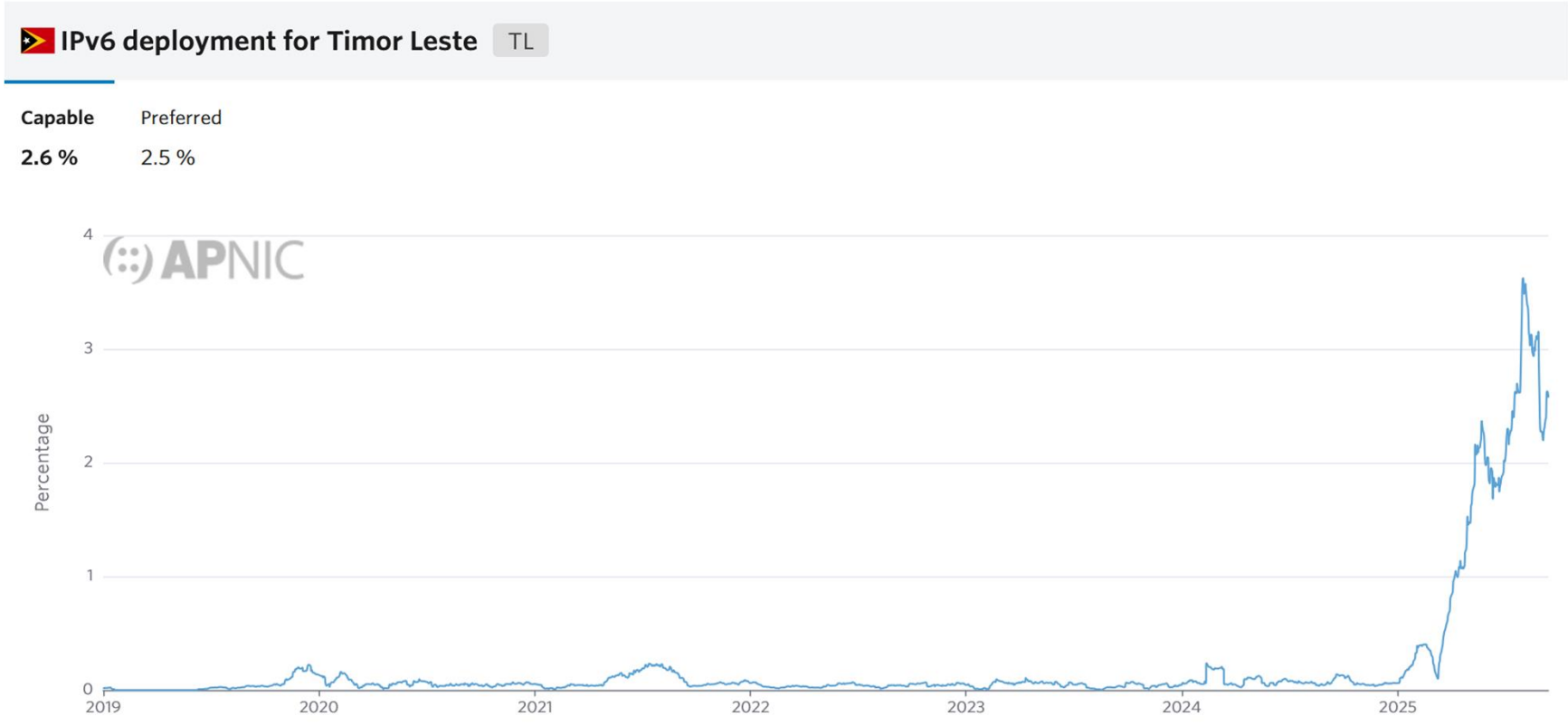
```
(Check on the query log)
$ sudo tail /var/cache/bind/queries
$ sudo tail /var/cache/bind/query-errors
```

RPKI: Where are we now?



https://rex.apnic.net/statistics?economy=TL

IPv6: Where are we now?



https://rex.apnic.net/statistics?economy=TL

Reference Links

Routinator

- NLnetLabs/routinator: An RPKI Validator - https://github.com/NLnetLabs/routinator

RPKI

- APNIC Academy - https://academy.apnic.net/en/webinar-courses/rpki-deployment

464XLAT

- NANOG 61 - https://archive.nanog.org/sites/default/files/wednesday_general_byrne_breakingfree_11.pdf
- APNIC 40 - https://conference.apnic.net/data/40/20150903_apnic40_464xlat_tutorial_mawatari_1441333278.pdf
- LACNIC 32 - https://www.lacnic.net/innovaportal/file/4016/1/464xlat-nat64-deployment-guidelines-v2.pdf

Thank You!