Mitigating Route Hijacking using RPKI and Automation

Aditya Kaul & Nitin Vig
Juniper Networks
July 2021
ABOUT US

Aditya Kaul  
kaula@juniper.net
- Architect @ Juniper Networks APAC
- 20 years in the Service Provider domain
- Keen interest in SR, SRv6 & Network Architecture Design

📍 Singapore

Nitin Vig  
nitinvig@juniper.net
- Architect @ Juniper Networks APAC
- 20 years in the Service Provider domain
- Keen interest in 5G, Edge cloud & Automation

📍 Malaysia
The Perfect world

Innocent user

198.51.0.10

INTERNET

198.51.0.0/22
AS64496

Service Provider
Prefix filters, IRR filtering, Peer lock, etc. are all In place?

- Prefix filters
- Peer lock aka “bignetworks filter”
- Bogon ASN filtering
- Bogon Prefix filtering
- Filter long ASN path
- Filter small prefixes
- IRR filtering
The Perfect world...or not (yet)?

I know all my customers and peers (all friends) and have filters and strict IRR applied.

However,...

- Prefix filters don’t care about the originating ASN or AS-PATH
- Peer Lock doesn’t cover every network and is arbitrary
- Downstream customers might use private ASN
- Downstream customers who are multihomed might unknowingly leak routes which they don’t originate
- IRR databases are far from correct, are incomplete or contain outdated data
IRR database accuracy

RIPE IRR

RADB IRR
BGP Hijacking is happening

April 2021 - Vodafone Idea (AS55410)
- AS55410 mistakenly announced over 30,000 BGP prefixes causing a 13x spike in inbound traffic to their network.
- VIL-AS-AP (Vodafone Idea) - hijacked 37739 prefixes - countries affected 164 - ASNs affected 4012 - duration 1:30:00
- Incident lasted around two hours. Users suffered slow connections and denial of service to some servers.

Source: https://www.manrs.org/2021/04/a-major-bgp-hijack-by-as55410-vodafone-idea-ltd/
Source: https://anuragbhatia.com/2021/04/networking/isp-column/large-prefix-hijack-from-vodafone-as55410/

April 2020 - Akamai, Amazon and Alibaba
- A massive BGP hijack involving over 8,800 prefixes affected companies such as Akamai, Amazon and Alibaba on April 1, 2020.
- Initiated by a Rostelecom user, the attack caused service disruptions throughout the world.
- Stricter network filtering by Rostelecom could have prevented the attack.

September 2020 - Telstra
- 500 prefixes wrongfully advertised as belonging to Telstra caused lengthy data detours.
- Incident was caused by post verification testing to address an unrelated software bug.

Source: https://www.anapaya.net/blog/border-gateway-protocol-hijacking-examples-and-solutions
What happened to our innocent user?

198.51.0.10

INTERNET

198.51.0.0/22
AS64496

Service Provider

198.51.0.0/23
AS65536

Malicious actor

Innocent user
So now what?

Origin Validation using RPKI

Resource Public Key Infrastructure (RPKI) is a method of cryptographic signing records that associate a prefix with an originating AS number.

All the five RIRs (AFRINIC, APNIC, ARIN, LACNIC & RIPE) provide a method for members to take a prefix/ASN pair and sign those with a ROA (Route Origin Authorization) record.

The ROA can then be used by operators to validate route advertisements. They are sure a route advertisement is intended by the legitimate owner.

Photo by Markus Spiske on Unsplash
Origin validation explained

1. The owner of a prefix registers with an RIR and creates a signed Validated ROA Payload (VRP)
2. RPKI validator downloads signed VRPs and verifies it
3. RPKI validator sends VRP to border routers that validate the BGP routes
RPKI Validator implementations


Some notable mentions:

**Routinator**: Project from NLnet Labs
[https://github.com/NLnetLabs/routinator](https://github.com/NLnetLabs/routinator)

**Fort Validator**: Part of the FORT routing security initiative by LACNIC and NIC.MX
[https://github.com/NICMx/FORT-validator](https://github.com/NICMx/FORT-validator)

**OctoRPKI**: Project from Cloud Flare
[https://github.com/cloudflare/cfrpki#octorpki](https://github.com/cloudflare/cfrpki#octorpki)

**Prover**: [https://github.com/lolepezy/rpki-prover](https://github.com/lolepezy/rpki-prover)
Perfect world routing

Route advertisement
- 198.51.0.0/22
- AS64496

List containing:
- valid prefix: 198.51.0.0
- origin ASN: AS64496
- expected mask length: /22

Validator → Router

Happy user being able to connect to right IP address
Even better world, Origin validation implemented

List containing:
- valid prefix: 198.51.0.0
- origin ASN: AS64496
- expected mask length: /22

Malicious Route advertisement
- 198.51.0.0/23
- AS65536

Origin Validation implemented

Malicious route not installed.
User will not connect to network advertising invalid prefix.
User will still connect to correct IP address.
But...only if the world was perfect

RPKI-ROV analysis of Unique Prefix-Origin Pairs (IPv4)

July 2019
(Valid: ~15%)

July 2020
(Valid: ~22%)

July 2021
(Valid: ~30%)

Source: https://rpki-monitor.antd.nist.gov
Protecting your network in an imperfect world (1/2)

Innocent user

198.51.0.10

Service Provider B

INTERNET

198.51.0.0/22
AS64496

Service Provider
Origin Validation implemented

198.51.0.0/23
AS65536

Malicious actor

Origin Validation NOT implemented

© 2021 Juniper Networks
Protecting your network in an imperfect world (2/2)

Innocent user

Service Provider B

Origin Validation NOT implemented

Service Provider

Origin Validation implemented

Someone hijacked my route

Malicious actor

198.51.0.0/24
198.51.1.0/24
198.51.0.0/22

AS64496

198.51.0.10

INTERNET
3 steps to detect & mitigate route hijacks with Automation

1. Identify your prefixes (‘source of truth’)
   - Create a prefix-list on the router (Manual or Automated)

2. Identify the hijack (if it occurs)
   - Compare received routes with the ‘source of truth’

3. Mitigate the attack
   - Advertise a more specific route to the router
3 steps to detect & mitigate route hijacks with Automation

1. Identify your prefixes ('source of truth')

2. Create a prefix-list on the router (Manual or Automated)

```conf
policy-options {
    prefix-list AS64496-prefixes {
        10.0.16.0/21;
        10.0.24.0/21;
        10.0.25.0/24;
        10.0.32.0/19;
        :
        :
        198.51.0.0/22;
    }
}
```

Note: Sample outputs from Juniper JUNOS
3 steps to detect & mitigate route hijacks with Automation

1. Identify your prefixes (source of truth)

Create a prefix-list on the router (Manual or Automated)

Router

```bash
policy-options {
  policy-statement hijack-check {
    term invalid-myprefixes {
      from {
        protocol bgp;
        prefix-list-filter AS64496-prefixes orlonger;
      }
      then {
        community add HIJACKED;
        reject;
      }
    }
    community HIJACKED members 64496:1111;
    community MITIGATED members 100:2222;
    community MYCUSTOMERS members 100:9999;
    community MYROUTES members 100:1000;
  }
}
```

```bash
protocols {
  bgp {
    group ebgp {
      type external;
      import hijack-check;
      family inet {
        unicast;
      }
      export [ EXPORT-PEER deny-all ];
      neighbor 17.8.2.2 {
        peer-as 200;
      }
      neighbor 17.8.3.2 {
        peer-as 300;
      }
    }
    neighbor 17.8.2.3 {
      peer-as 400;
    }
  }
}
```

Note:
- Sample outputs from Juniper JUNOS
- Ensure proper policy hierarchy to ensure routes are not dropped
- Additional consideration required for ASN hijacking case
3 steps to detect & mitigate route hijacks with Automation

2

Identify the hijack (if it occurs)

Compare received routes with the prefix-list

root@isp-r1> show route community 64496:1111 all

inet.0: 781928 destinations, 2345613 routes (781922 active, 0 holddown, 781864 hidden)
+ = Active Route, - = Last Active, * = Both

198.51.0.0/23 [BGP ] 00:01:17, localpref 100
AS path: 200 65535 I, validation-state: unverified
> to 17.8.2.2 via ge-0/0/2.0

Note: Sample outputs from Juniper JUNOS
3 steps to detect & mitigate route hijacks with Automation

1. Identify the hijack (if it occurs)
2. Compare received routes with the prefix-list
3. Paragon Insights (Automation system)

Identify the hijack (if it occurs)

Compare received routes with the prefix-list

Paragon Insights (Automation system)

Telemetry/Netconf/SNMP

Provider network

© 2021 Juniper Networks
3 steps to detect & mitigate route hijacks with Automation

1. Advertise a more specific route to the router

   root@isp-r1# show groups __automation_routeHijack__
   routing-options {
     static {
       route 198.51.0.0/24 {
         discard;
         no-install;
         community 64496:2222;
       }
       route 198.51.1.0/24 {
         discard;
         no-install;
         community 64496:2222;
       }
     }
   }

2. Mitigate the attack

   Pushing more-specific route

   Paragon Insights
   (Automation system)

   Netconf/CLI/OpenConfig

3. Automation system

   Note: Sample outputs from Juniper JUNOS
Ready? Call to action!

To Do:
- Sign your Prefixes (create ROAs)
- Setup a Validator
- Configure your routers
- Use automation where relevant
- Support work in IETF and APNIC

Start now: make the internet more reliable and secure!
Thank You
COMMON TYPES OF ROUTE HIJACKS

<table>
<thead>
<tr>
<th>Hijack type</th>
<th>Impact to the ISP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1  Someone hijacks my route</td>
<td>Traffic destined to me is blackholed</td>
</tr>
<tr>
<td>2  Me or my customer inadvertently hijack someone’s route</td>
<td>My network becomes a sink for the hijacked route</td>
</tr>
<tr>
<td>3  Someone hijacks someone else’s route</td>
<td>I might potentially send traffic to the wrong destination</td>
</tr>
</tbody>
</table>