NTT Peer Locking

Deployment of NTT "Peer Locking" route leak prevention mechanism.

Date : Sat Apr 30 00:00:51 PDT 2016

Policy : NTT Global Peering peering@ntt.net

Operational: NTT Global NOC noc@ntt.net

Audience : ExamplePeeringNetwork / AS ExamplePeerASN

This document describes NTT's Peer Locking mechanism, an effort to increase the Internet's routing security by protecting NTT's BGP neighbors with an additional layer of filtering.

NTT seeks your explicit permission to deploy certain as-path filters inside AS 2914. If you agree, the risk of your prefixes being accepted via unauthorised paths during route leak events will be significantly reduced on a global scale. You are encouraged to replicate this mechanism or a variant in your own network!

There are no fees associated with Peer Locking, no contractual or legal ramifications. Your network being designated with `Protected ASN` status is entirely optional and you as peering partner may request NTT to change this status at any time. No operational impact of any nature is expected should you permit NTT to deploy this additional layer of protection.

Peer Locking relies on peering partners communicating to NTT which networks are authorised upstreams, or declaring that they have no upstreams at all. NTT's implementation allows very fast network-wide updates should the situation demand swift action. The implementation offers granularity up to the continent level: it is no issue if a peering partner is transit-free in one region but not in others.

Technical details:

Terminology:

- **`Protected ASN`** - this is your ASN (AS ExamplePeerASN), all prefixes which contain a `Protected ASN` in the AS_PATH but are received via an unauthorised eBGP neighbor will be rejected.
- **`Allowed Upstream`** - a `Protected ASN` may indicate to NTT that a certain ASN is allowed to propagate prefixes which contain the `Protected ASN` in the AS_PATH. `Allowed Upstreams` can be configured globally, per region or set to "None".

Both `Protected ASNs` and `Allowed Upstreams` must be directly connected to NTT AS2914 backbone in multiple regions to be considered eligible for either of the two roles.

Default operating mode

When a peering partner agrees to be elevated to the status of `Protected ASN` (by default) NTT will only accept prefixes which contain the Protected ASN in the AS_PATH if they are received over the direct BGP sessions between NTT and the `Protected ASN`. In most cases, especially with larger peering partners, this default operating mode is sufficient.

In some cases a peering partner might want NTT to accept prefixes via an intermediate network. NTT needs to be made aware of such cases. Peers need to proactively communicate who their `Allowed Upstreams` are.

Example case with two eBGP neighbors:

Consider the following (hypothetical) eBGP neighbors of NTT AS 2914:

- AS 267 (AS-NETHERNET) a transit customer of NTT.
- AS 15562 (AS-SNIJDERS) is a transit customer of NTT.
- AS 8283 (AS-COLOCLUE) is a peering partner of NTT.
- AS ExamplePeerASN (ExamplePeeringNetwork) is a peering partner of NTT.

In this example both AS 8283 and AS ExamplePeerASN have given NTT permission to enable Peer Locking, both AS 8283 and AS ExamplePeerASN are a `Protected ASN`.

If these four eBGP neighbors are configured on the same router, the following example JunOS configuration would be generated by NTT's network orchestration software. Please note that this an incomplete configuration, in reality the configs are far more elaborate!

It is important to note that NTT has configured Peer Locking filters on each and every eBGP session in its entire network, no exceptions. Peer Locking is address family agnostic: the **same** as-path filter is applied on **both** the `IPv4` and `IPv6` sessions with an adjacent network.

JunOS Example #1

```
as-path lock-AS267-in ".* (8283|ExamplePeerASN) .* ";
as-path lock-AS8283-in ".* (ExamplePeerASN) .* ";
as-path lock-AS15562-in ".* (8283|ExamplePeerASN) .* ";
as-path lock-ASExamplePeerASN-in ".* (8283) .* ";

policy-statement AS267-in-v6 {
  term blockasns {
    from as-path lock-AS267-in;
    then reject;
  }
  term other-stuff {
    .. mark as customer ..
    next policy;
  }
}

policy-statement AS15562-in-v6 {
  term blockasns {
    from as-path lock-AS15562-in;
    then reject;
  }
  term other-stuff {
    .. mark as customer ..
    next policy;
  }
}

policy-statement AS8283-in-v6 {
  term blockasns {
    from as-path lock-AS8283-in;
     then reject;
```

```
  }
  term other-stuff {
    .. mark as peer ..
    next policy;
  }
}

policy-statement ASExamplePeerASN-in-v6 {
  term blockasns {
    from as-path lock-ASExamplePeerASN-in;
    then reject;
  }
  term other-stuff {
    .. mark as peer ..
    next policy;
  }
}

protocols {
  bgp {
    advertise-peer-as;
    log-updown;
    group eBGP-customers-v6 {
      neighbor 2001:db8:1::1 {
        description "NETHER";
        import [ reject-bogon-v6 AS267-in-v6 final-filter ];
        family inet6 {
          unicast {
            prefix-limit {
              maximum 10;
              teardown 90 idle-timeout 10;
            }
          }
        }
        export [ ... ];
        peer-as 267;
      }
      neighbor 2001:db8:2::1 {
        description "SNIJDERS";
        import [ reject-bogon-v6 AS15562-in-v6 final-filter ];
        family inet6 {
          unicast {
            prefix-limit {
              maximum 10;
              teardown 90 idle-timeout 10;
            }
          }
        }
        export [ ... ];
        peer-as 15562;
      }
    }
    group eBGP-peers-v6 {
      neighbor 2001:db8:3::1 {
        description "COLOCLUE";
        import [ reject-bogon-v6 AS8283-in-v6 final-filter ];
        family inet6 {
          unicast {
            prefix-limit {
              maximum 10;
```

```
            teardown 90 idle-timeout 10;
          }
        }
      }
      export [ ... ];
      peer-as 8283;
    }
    neighbor 2001:db8:4::1 {
      description "ExamplePeeringNetwork";
      import [ reject-bogon-v6 ASExamplePeerASN-in-v6 final-
filter ];
      family inet6 {
        unicast {
          prefix-limit {
            maximum 81213;
            teardown 90 idle-timeout 10;
          }
        }
      }
      export [ ... ];
      peer-as ExamplePeerASN;
    }
  }
 }
}
```

JunOS example #2 (allowed upstream)

Given example #1, at some point AS 8283 might communicate to NTT that for whatever reason, 8283 will be using 15562 as upstream (temporary or permanent does not matter). At that point AS 15562 would become an `Allowed Upstream` for `Protected ASN` 8283. Note that 15562 is not part of this communication, this is exclusively between NTT and 8283.

As soon as the NTT NOC inputs the requested change into the NTT network orchestration backend, the only part about the generated configuration that would become different are the as-path filters:

```
as-path lock-AS267-in ".* (8283|ExamplePeerASN) .* ";
as-path lock-AS8283-in ".* (ExamplePeerASN) .* ";
as-path lock-AS15562-in ".* (ExamplePeerASN) .* ";
as-path lock-ASExamplePeerASN-in ".* (8283) .* ";
```

As you can see in the above configuration, combined with the earlier policy-statements, the resulting policy would allow paths such as ^8283$ and `^15562_8283$` - but not `^267_8283_....$`

Communicating changes to NTT

Any change requests related to Peer Locking can be emailed to the NTT NOC at noc@ntt.net. Our NOC will assign the request an ID for tracking purposes and work with NTT's engineering

department to review the requested change. We strive to resolve Peer Locking change requests within 24 hours.

Conclusion

NTT believes that the Peer Locking mechanism, when applied to the twenty largest networks in the world, will greatly reduce the impact and spread of routeleaks.

NTT actively monitors the default-free zone through tools such as https://puck.nether.net/bgp/leakinfo.cgi and we have already noticed a vast improvement for networks that agreed to be a `Protected ASN`.

Feel free to contact NTT should you have any questions.