ROUTE FILTERING:

If you accept free beer, expect an hangover

Thomas Mangin
AS30740

Linx 57, 21st of may 2007

Why I am standing here ?

- First hand experience with a Leak two weeks ago
  - Sending a full routing table all my linx peers
  - And transit providers
  - Since on more leak was announced on the Linx mailing-list
  - If you dig the linx-ops archive, leak is an industry sport

- Maxed our linx port quickly ..
  - Traffic mainly going to Level3
  - Surely local-preference to blame
  - ... and peers

What should have happened

- Nothing should have happened
  - I should have not leaked, but I did ..
  - Transit providers handled the leak gracefully thanks to filters, as did some peers
- And then NOTHING
  - The routes should have been filtered inbound
  - Or max-prefix should have killed the session
- Two layers of protection failed, or were absent, on both side !

Protect yourself and your peers

Protect your peers from your leak or be ready ...

- John will have do a repeat of this talk
- Worse, you will spend days with sessions in active (shameless plug to 31 peers out of 138)

Protect yourself.

- I will do some maintenance work tomorrow night
- Nobody's perfect ..

It only takes a few hours to prevent further leaks.

- If I done it ....

Protect your peers : Identify your transit routes

Routes can be identified by and manipulated using:

(1) outbound as-path filtering
refuse to announce your transit

(2) community to decide what to announce

(3) prefixes :
refuse announce bogons
refuse announce small prefixes
refuse announce known range (ix, etc.)

(4) do not propagate private-asn

as-path filtering -outbound

```
/* define the routes we have learned from transit (example) */
as-path routes-level3 3356.*;
as-path routes-sprint 1239.*;

/* create a policy blocking their distribution */
[edit policy-options]
policy-statement no-transit {
    term remove-path {
        from {
            protocol bgp;
            as-path [ routes-level3 route-sprint ];
        }
        then reject;
    }
}

/* make sure that no linx peer will ever get them again */
[edit protocols bgp group linx]
export [ no-transit ];
```

community tagging

```
/* define a communtiy to identify routes learned from transit */
community route-transit members 174:1239;

/* create a policy to apply this community to a route */
policy-statement tag-transit {
    then {
        community add route-transit;
    }
}

/* make sure all routes from transit have that community */
[edit protocols bgp group transit]
import [ tag-transit tag-transit-provider-specific ];

(repeat with peers)
```

community filtering

```
/* define a policy rejecting routes identified as transit */
[edit policy-options]
policy-statement export-transit {
    term remove-peering {
        from {
            protocol bgp;
            community route-transit;
        }
        then reject;
    }
    term remove-peering ...
    term remove-community ...
    term prepend-one-time ...
}

/* and make sure no linx peer sees it */
[edit protocols bgp group linx]
export [ export-peering export-linx ];
```

Do not do typo with your community definition without (1).. it hurts ..

Those route no-one should have

```
/* match and refuse any route smaller/longer than a /24 */

[edit policy-options]
policy-statement no-small-prefixes {
    from {
        route-filter 0.0.0.0/0 prefix-length-range /25-/32 reject;
    }
    then reject;
}

/* bogon, rfc1819, etc. */

[edit policy-options]
policy-statement no-bogons {
    from {
        route-filter 224.0.0.0/4 orlonger reject;
        ......
    }
}
```

Those routes no-one should have

```
/* Linx LAN */

policy-statement no-ix {
    from {
        route-filter 195.66.224.0/22 orlonger reject;
    }
    then reject;
}

/* should never get in or out */

[edit protocols bgp group linx]
export [ no-small-prefixes no-ix no-bogons ];
import [ no-small-prefixes no-ix no-bogons ];
```

Protect yourself

Do not trust your peer /mainly/ if you are peering with me:

(1) use max-prefix to limit the number of route a peer can send you
it is cheap, fast and works

(2) refuse route with an as-path ..
of a “T1” coming from peers
of your customers from peer or transit
with reserved ASN

(3) Create inbound route filters

Max-Prefix

Max-prefix will shutdown a session should the ebgp speaker send you more than a predefined number of routes (was it necessary to say it ?)

```
neighbor 195.66.224.xxx {
    description "AS-ACCEPTED | Peer name | noc@peer.co.uk | AS-SENT";
    family inet {
        unicast {
            prefix-limit {
                maximum 150;
                teardown 80;
            }
        }
    }
    peer-as 1234;
}
```

as-path filtering -outbound

```
/* define the routes you will never see through peers */
as-path leaked-sprint ".{1,}1239.*";
as-path leaked-telia ".{1,}1299.*";

/* create a policy blocking their distribution */
[edit policy-options]
policy-statement no-leak {
    term remove-path {
        from {
            protocol bgp;
            as-path [ leaked-telia leaked-spring ];
        }
        then reject;
    }
}

/* make sure that no linx peer will ever get them again */
[edit protocols bgp group linx]
import [ no-leak ];
```

as-path filtering outbound

Limitations :

On cisco this works great as the count is performed on prefix accepted

On juniper not as good the counting is done on prefix received (before any kind of filtering)

Please push for this feature request to your SE :
http://juniper.cluepon.net/index.php/ER_BGP_Prefix_limit_enhancements

Filter on IRR DB

Some tools exist to help with the generation of filter based on the content of the IRR DB (RIPE, ARIN, etc.)

http://irrpt.sourceforce.net/
Gather and Track prefix within AS-Macro.

Conclusion

Renesys BGP white paper,
Everything I said but in plain understandable english

http://www.renesys.com/tech/white_papers/WP_BGP_rev6.pdf

For those of you who prefer cisco talks
http://www.apnic.org/meetings/15/tutorials/index.html