Routing in Fat Trees (RIFT)

Update

IETF 105

Tony Przygienda, Editor, Juniper
All RIFT routers MUST support IPv4 forwarding and MAY support IPv6 forwarding. A three way adjacency over IPv6 addresses implies support for IPv4 forwarding. A node that does not process received IPv6 LKE MUST NOT originate IPv6 LKEs.

5.4.3. Security Envelope

RIFT MUST be carried in a mandatory secure envelope illustrated in Figure 31. Local configuration can allow to skip the checking of the envelope’s integrity.

Any implementation including RIFT security MUST generate and wrap around local nonces properly. All implementation MUST reflect the neighbor’s nonces. An Implementation SHOULD increment a chosen nonce on every LIE FSM transition that ends up in a different state from

RIFT MUST be carried in a mandatory secure envelope illustrated in Figure 31. Any value in the packet following a security fingerprint MUST be used only after the according fingerprint has been validated.

Any implementation including RIFT security MUST generate and wrap around local nonces properly. When a nonce increment leads to `undefined nonce’ value the value SHOULD be incremented again immediately. All implementation MUST reflect the neighbor’s nonces.

As optional optimisation, an implemenation MAY send one LIE with previously negotiated neighbor’s nonce to try to speed up a neighbor’s transition from 3-way to 1-way and MUST revert to sending `undefined_nonce’ after that.
Published -06, Repo -07 Changes vs. -05

5.4.5. Lifetime

Protecting lifetime on flooding may lead to excessive number of security fingerprint computation and hence an application generating such fingerprints on TIES SHOULD round the value down to the next `rounddown_lifetime_interval` defined in the scheme when sending TIES.

```c
struct LinkIDPair {
    // ** node-wide unique value for the local link */
    1: required common.LinkIDType local_id;
    // ** received remote link ID for this link */
    2: required common.LinkIDType remote_id;

    // ** optionally describes the local interface index of the link */
    10: optional common.PlatformInterfaceIndex platform_interface_index;
    // ** optionally describes the local interface name */
    11: optional string platform_interface_name;

    // ** optional indication whether the link is secured, i.e. protected by outer key, absence of this element means no indication, undefined outer key means not secured */
    12: optional common.OuterSecurityKeyID trusted_outer_security_key;
}
```

```c
struct TIESHeaderWithLifetime {
    1: required TIEHeader header;
    // ** remaining lifetime that expires down to 0 just like in ISIS. TIES with lifetimes differing by less than `lifetime_diffIgnore` MUST be considered EQUAL. */
    2: required common.LifetimeInSecType remaining_lifetime;
}
```

```c
struct PrefixAttributes {
    2: required commonMetricType metric = common.default_distance;
    // ** generic unordered set of route tags, can be redistributed to other protocols or used within the context of real time analytics */
    3: optional <common.HostType> tags;
    // ** optional monotonic clock for mobile addresses */
    4: optional common.MonotonicClock monotonic_clock;
    // ** optionally indicates the interface is a node loopback */
    5: optional bool loopback = false;
    // ** indicates that the prefix is directly attached, i.e. should be routed to even if the node is in overload. */
    7: optional bool directly_attached = true;
}
```
Rounds of Juniper vs. Python-RIFT Interop with Security Envelope

- Python-RIFT has full single plane implementation now
- 0.11 Juniper about to be released (-07 draft implementation)
- 0.11 security envelope interop concluded
- 0.11 Juniper will be released to public
  - Will include specification of
    - Configuration API
    - Operational state API
    - Real-time analytics API
Early Directorate Review & Discussions

• Discussion on router requirements for RIFT
  • Phrasing requiring v4 forwarding support will be removed
  • Needs addition to LIE in schema to indicate whether v4 forwarding is supported

• Review came in
  • Mostly editorial/better wording

• Review concluded, results posted, will be published on -07 after IETF
THANK YOU FOR YOUR ATTENTION