graph_redhat

```
CREATE TABLE test.t_tmp_craw_redhat(id varchar, vul_msg jsonb)


CREATE TABLE test.t_tmp_node_vul_REDHAT AS
WITH wt_cwe AS
(
    SELECT tmp2.cve_id, jsonb_agg(tmp2.weakness_tmp)  AS weakness
    FROM
    (
        SELECT tmp1.cve_id, jsonb_build_object('type', NULL, 'cweId',  trim(trim(UNNEST(string_to_array(tmp1.cw
        FROM
        (
        SELECT UNNEST(string_to_array(cr.vul_msg->> 'cwe','->')) cwes, cr.vul_msg ->> 'name' AS cve_id
        FROM test.t_tmp_craw_redhat cr
        )tmp1
    )tmp2
    GROUP BY tmp2.cve_id
),
wt_cvss_msg AS
(
  SELECT ('{"cvssV2_0":{"version":"2.0", "baseScore":"'|| COALESCE( (t.vul_msg -> 'cvss' ->> 'cvss_base_score')
                       "vectorString":"' || COALESCE( (t.vul_msg -> 'cvss' ->> 'cvss_scoring_vector'),'')  || '"
                       "baseSeverity":null, "attackVector":null,
                      "confidentialityImpact":null,
                      "integrityImpact":null, "availabilityImpact":null, "userInteraction":null, "attackComplexi
                     },
         "cvssV' || COALESCE( REPLACE(t.cvss3_version,'.','_'), '3_0') || '":{"version":"' || COALESCE( t.cvss3
                      "baseScore":"'|| COALESCE( (t.vul_msg -> 'cvss3' ->> 'cvss3_base_score'), '') || '",
                      "vectorString":"' || COALESCE( (t.vul_msg -> 'cvss3' ->> 'cvss3_scoring_vector'), '')  ||
                      "baseSeverity":null,
                      "attackVector":null,
                      "confidentialityImpact":null,
                      "integrityImpact":null, "availabilityImpact":null, "userInteraction":null, "attackComplexi
                     }
       }')::jsonb AS severity,
--       t.vul_msg -> 'cvss' AS cvss2,
--       t.vul_msg -> 'cvss3' AS cvss3
       t.vul_msg ->> 'name' AS cve_id
    FROM
    (
        SELECT cr.vul_msg,  split_part( split_part( cr.vul_msg -> 'cvss3' ->> 'cvss3_scoring_vector', '/', 1),
        FROM test.t_tmp_craw_redhat cr
    )t
)
SELECT cr.vul_msg ->> 'name' AS id,
       '[]' AS aliases,
       jsonb_build_object('discovery', NULL, 'identifier', null) AS SOURCE,
       jsonb_build_object('title', cr.vul_msg ->> 'statement', 'details', cr.vul_msg ->> 'details') AS descript
       COALESCE(wc.weakness,'{}') AS weakness,
```

```
       COALESCE(cm.severity, '{}') AS severity,
       jsonb_build_object('published', cr.vul_msg ->> 'public_date', 'lastModified', NULL, 'datePublic', NULL)
       ('{"impact_info":{"impacts":null, "impactScore":null},
            "exploit_info":{"exploitable":null, "exploits":null, "exploit_url":null, "exploitabilityScore":null
            "PoC_info":{"PoC_available":null, "PoC":null, "PoC_url":null},
            "patch_info":{"patch_available":null, "patch_url":null},
            "report_status":null}')::jsonb ||
        jsonb_build_object('solution_info', cr.vul_msg -> 'mitigation' ->> 'value')
            AS vul_status
FROM test.t_tmp_craw_redhat cr
LEFT JOIN wt_cwe wc
       ON (cr.vul_msg ->> 'name') = wc.cve_id
LEFT JOIN wt_cvss_msg cm
       ON (cr.vul_msg ->> 'name') = cm.cve_id;

--vul node
DELETE FROM test.dws_graph_node_vul WHERE vul_source = 'REDHAT';
ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
INSERT INTO test.dws_graph_node_vul
SELECT nextval('cve_graph_seq') AS seq, tmp.*
FROM
(
SELECT  DISTINCT  id ,  aliases , "source", description ,weakness, severity, time_info,  vul_status, 'REDHAT' A
)tmp ;

SELECT *FROM test.t_tmp_node_vul_REDHAT LIMIT 10;

--affected component
DROP TABLE test.t_tmp_graph_node_component_redhat ;
CREATE TABLE test.t_tmp_graph_node_component_redhat AS
WITH wt_upstream_fix AS
(
    SELECT tmp2.MODULE,  tmp2.cve_id, '[' || string_agg(DISTINCT tmp2.version_num, ',') || ']' AS upstream_fix_
    FROM
    (
        SELECT split_part(tmp1.mig, ' ', tmp1.blank_count + 1) AS version_num,
              trim(replace( tmp1.mig, split_part(tmp1.mig, ' ', tmp1.blank_count + 1), '')) AS module,
              tmp1.*
        FROM
        (
            SELECT ( length(t.mig) - length(replace(t.mig,' ','')) ) AS blank_count, t.*
            FROM
            (
                SELECT trim( UNNEST(string_to_array(
                            trim( UNNEST( string_to_array( cr.vul_msg ->> 'upstream_fix' , ',') )), 'and') )) A
                            cr.vul_msg ->> 'name' AS cve_id
                FROM test.t_tmp_craw_redhat cr
            )t
            WHERE t.mig IS NOT NULL
        )tmp1
    )tmp2
    GROUP BY tmp2.MODULE, tmp2.cve_id
),
wt_package_state AS
(
    SELECT jsonb_build_object('release', tmp1.product_name, 'fixed_version', uf.upstream_fix_version, 'urgency'
           tmp1.product_name, tmp1.package_name, tmp1.cve_id
    FROM (
```

```
        SELECT DISTINCT package_state ->> 'fix_state' AS fix_state, --AS urgency
               package_state ->> 'package_name' AS package_name, -- AS MODULE
               package_state ->> 'product_name' AS product_name, -- AS product
               t.vul_msg ->> 'name' AS cve_id
            FROM test.t_tmp_craw_redhat t,
            jsonb_array_elements(t.vul_msg -> 'package_state') package_state
      )tmp1
   LEFT JOIN wt_upstream_fix uf
          ON tmp1.package_name = uf.MODULE
          AND tmp1.cve_id = uf.cve_id
),
wt_affected_release AS
(
    SELECT 'RedHat' AS vendor,
            COALESCE(affected_release ->> 'package','') AS package_with_version, --AS version
            COALESCE (affected_release ->> 'product_name', '*') AS product_name, --AS release
            regexp_replace(affected_release ->> 'package', '^(.*?)-[0-9]+:([0-9]+\.[0-9]+\.[0-9]+).*$', '\1') A
            regexp_replace(affected_release ->> 'package', '^(.*?)-[0-9]+:([0-9]+\.[0-9]+\.[0-9]+).*$', '\2') A
            '*' AS ecosystem,
            t.vul_msg ->> 'name' AS cve_id
    FROM test.t_tmp_craw_redhat t,
    jsonb_array_elements(t.vul_msg -> 'affected_release') affected_release
)
SELECT ar.cve_id, ar.vendor, ar.ecosystem, ar.product_name, ar.module_name, COALESCE(ps.fixed_versions, '[]',
       jsonb_build_object('release', ar.product_name, 'version', COALESCE(uf.upstream_fix_version, ar.package_w
FROM (SELECT *FROM wt_affected_release WHERE module_name IS NOT NULL) ar  --过滤掉没有package的数据
LEFT JOIN wt_upstream_fix uf
       ON ar.module_name = uf.MODULE
       AND ar.cve_id = uf.cve_id
LEFT JOIN wt_package_state ps
       ON ar.product_name = ps.product_name
       AND ar.module_name = ps.package_name
       AND ar.cve_id = ps.cve_id ;


DELETE FROM test.dws_graph_node_affected_component WHERE vul_source = 'REDHAT';
ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
INSERT INTO test.dws_graph_node_affected_component
  SELECT nextval('cve_graph_seq') AS seq, tmp.*
  FROM (
    SELECT   DISTINCT product_name, vendor, module_name, ecosystem, 'REDHAT' AS vul_source   FROM   test.t_tmp_
  )tmp;

SELECT *FROM test.t_tmp_graph_node_component_redhat LIMIT 100;

--affected component
  --seq int , vul_id , component_name , vendor , package_name , ecosystem , repo_url ,
  --platform , collectionUrl , defaultStatus , affected_versions ,  unaffected_versions , vul_source
DELETE FROM test.dws_graph_relationships_affected_components WHERE vul_source = 'REDHAT';
ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
INSERT INTO test.dws_graph_relationships_affected_components
  SELECT nextval('cve_graph_seq') AS seq, tmp.*
  FROM (
    SELECT   DISTINCT cve_id, product_name, vendor, module_name, ecosystem, '' repo_url, '' AS platform,
             '' AS collectionurl, '' AS defaultstatus, affected_versions, COALESCE(fixed_versions,'{}'), 'REDHA
  )tmp;

--reference
```

```
 CREATE TABLE test.t_tmp_graph_node_refs_REDHAT AS
SELECT t.vul_msg ->> 'name' AS cve_id,  UNNEST(string_to_array( refs ->> 0 , e'\n'))AS ref_url
FROM test.t_tmp_craw_redhat t,
jsonb_array_elements(t.vul_msg -> 'references') refs

--node seq int, url varchar, vul_source varchar
DELETE FROM test.dws_graph_node_refs WHERE vul_source = 'REDHAT';
ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
INSERT INTO test.dws_graph_node_refs
  SELECT nextval('cve_graph_seq') AS seq, tmp.*
  FROM (
    SELECT   DISTINCT ref_url, 'REDHAT' AS source  FROM   test.t_tmp_graph_node_refs_REDHAT
  )tmp;

--relationship seq int , vul_id varchar, url varchar, tags varchar, description varchar, vul_source varchar
DELETE FROM test.dws_graph_relationships_refs WHERE vul_source = 'REDHAT';
ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
INSERT INTO test.dws_graph_relationships_refs
  SELECT nextval('cve_graph_seq') AS seq, tmp.*
  FROM (
    SELECT   DISTINCT cve_id AS id, REf_url , '' AS tags, '' AS ref_desc , 'REDHAT' AS vul_source  FROM   test.
  )tmp;




--cwe
CREATE TABLE test.t_tmp_graph_node_cwe_REDHAT AS
SELECT tmp1.cve_id, trim(trim(UNNEST(string_to_array(tmp1.cwes, '|')),'('),')') AS cwe_id
FROM
(
SELECT UNNEST(string_to_array(cr.vul_msg->> 'cwe','->')) cwes, cr.vul_msg ->> 'name' AS cve_id
FROM test.t_tmp_craw_redhat cr
)tmp1

--cwe node
--seq , id , vul_source
DELETE FROM test.dws_graph_node_cwe WHERE vul_source = 'REDHAT';
ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
INSERT INTO test.dws_graph_node_cwe
  SELECT nextval('cve_graph_seq') AS seq, tmp.*
  FROM (
    SELECT   DISTINCT cwe_id, 'REDHAT'   FROM   test.t_tmp_graph_node_cwe_REDHAT
  )tmp;

--relationships cwe
-- seq int , vul_id varchar, cwe_id varchar, cwe_type varchar, cwe_desc varchar, vul_source varchar
DELETE FROM test.dws_graph_relationships_cwe WHERE vul_source = 'REDHAT';
ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
INSERT INTO test.dws_graph_relationships_cwe
  SELECT nextval('cve_graph_seq') AS seq, tmp.*
  FROM (
    SELECT   DISTINCT cve_id AS id, cwe_id , '' AS cwe_type, '' AS cwe_desc , 'REDHAT' AS vul_source  FROM   te
  )tmp;
```